A CLUSTER-BASED EXTERNAL PLAGIARISM AND PARALLEL CORPORA DETECTION METHOD

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING

AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Ceyhun Efe Karbeyaz

July, 2011

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Fazlı Can(Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Dr. Aybar Acar

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Mehmet Kalpaklı

Approved for the Graduate School of Engineering and Science:

Prof. Dr. Levent Onural
Director of the Graduate School of Engineering and Science

ABSTRACT

A CLUSTER-BASED EXTERNAL PLAGIARISM AND PARALLEL CORPORA DETECTION METHOD

Ceyhun Efe Karbeyaz
M.S. in Computer Engineering
Supervisor: Prof. Dr. Fazlı Can
July, 2011

Today different editions and translations of the same literary text can be found. Intuitively such translations that are based on the same literary text are expected to possess significantly similar structure. In the same way, it is possible that a text that is suspected to have plagiarism can possess structural similarities with the text that is believed to be the source of the plagiarism. Textual plagiarism implies the usage of an author's text, his/her work or the idea that is inserted in another textual work without giving a reference or without taking the permission of the original text's author. Today, existing intrinsic and external plagiarism detection methods tend to detect plagiarism cases within a given dataset in order to run these algorithms in a reasonable amount of time. Hence a reference document set is built in order to search for plagiarism cases successfully by these algorithms. In this thesis, a method for detecting and quantifying the external plagiarism and parallel corpora is introduced. For this purpose, we use the structural similarities in order to analyze plagiarism detection problem and to quantify the similarity between given texts. In this method, suspicious and source texts are partitioned into corresponding blocks. Each block is represented as a group of documents where a document consists of a fixed amount of words. Then, blocks are indexed and clustered by using the cover coefficient clustering algorithm. Cluster formations for both texts are then analyzed and their similarities are measured. The results over PAN'09 plagiarism dataset and over different versions of the famous literary text classic Leylâ and Mecnun show that the proposed method successfully detects and quantifies the structurally similar plagiarism cases and succeeds in detecting the parallel corpora.

Keywords: Plagiarism detection, parallel corpora detection, clustering.

ÖZET

KÜMELEMEYE DAYALI HARİCİ İNTİHAL VE PARALEL METİN TESPİT YÖNTEMİ

Ceyhun Efe Karbeyaz
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Prof. Dr. Fazlı Can
Temmuz, 2011

Günümüzde aynı edebi eserin farklı versiyonlarını detaylı bir aramayla bulabilmek mümkündür. Sezgisel olarak bu tür aynı kaynak tabanlı çeviri eserlerin birbirlerine benzer yapıda olmaları beklenmektedir. Aynı şekilde, intihal şüphesi taşıyan bir yazı metnin, intihal yapılan orijinal eser ile de yapısal olarak benzemesi olasıdır. Yazısal intihal ile kastedilen, bir yazarın yazdığı herhangi bir metninin, üslubunun veya belirttiği fikrin, yazar lehine kaynak gösterilmeden başka biri tarafından yazarın onayını almadan kullanılmasıdır. Günümüzdeki içsel ve harici yazısal intihal tespit yöntemleri var olan intihalin tespitini makul zaman dilimleri içerisinde sonuçlandırabilmek için yapılan yazısal intihalin kapsamını sınırlandırma yoluna gitmişler ve intihali arayabilmenin önkoşulu olarak bir referans doküman kümesine ihtiyaç duymuşlardır. Bu da intihal tespit yönteminde referans doküman kümesinin başarıyla oluşturulması gibi başka sorunların varlığını ortaya koymuştur. Bu tez çalışmasında bir harici intihal ve benzer yapı tespit ve ölçme yöntemi önerilmiştir. İntihal tespit problemini analiz etmek ve benzerligi ölçmek için metinlerdeki yapısal benzerlikten faydanılmıştır. Bu yöntem dahilinde öncelikle şüpheli ve kaynak metinler karşılıklı bloklara bölünmüştür. Oluşturulan her bir blok sabit sayıda kelime içeren bir grup dökümandan oluşmaktadır. Daha sonra bloklar indekslenmiş ve kapsama katsayısına dayalı kümeleme yöntemiyle kümelenmiştir. Her iki metnin oluşan küme yapıları incelenmiş ve benzerlikleri ölçülmüştür. PAN'09 intihal veri kümesi ve ünlü edebi eser Leylâ ve Mecnun'un farklı versiyonları üzerinde yapılan test sonuçlarına göre önerilen yöntem benzer yapı tespitini ve yapısal olarak benzerlik gösteren intihal durumlarını başarıyla tespit edebilmektedir.

Anahtar sözcükler: İntihal tespiti, benzer yapı tespiti, kümeleme.

Acknowledgement

I would like thank my advisor Prof. Dr. Fazlı Can, who helped me to reach my goals that I was dreaming about six years ago, for his helpful pointers about academic and non-academic life. I am sure if I did not have his guidance during my graduate studies, my life weren't improved that much completely and deeply.

I am grateful to the members of the jury, Dr. Aybar Acar and Asst. Prof. Dr. Mehmet Kalpaklı for reading and reviewing my thesis.

I would like to thank to my family, especially to my brother Ersel Karbeyaz and sister-in-law Başak Ülker Karbeyaz for their assistance during my undergraduate and graduate studies.

I would like to thank to members of Bilkent Information Retrieval Group, especially to Ethem Fatih Can and Cem Aksoy for their assistance during my graduate studies.

I would like to acknowledge Scientific and Technical Research Council of Turkey (TÜBITAK) for their support under the grant number 109E006. I also would like to thank to Bilkent Computer Engineering department for their financial support during my studies and conference visits.

I am also grateful to my friends Ahmet Yeniçağ, Anıl Türel, Çağrı Toraman and Emir Gülümser for their friendship and assistance during my studies.

to my mother and father. . .

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivations

Today different translations of the same literary text can be found. Intuitively such translations that are based on the same literary text are expected to possess significantly similar structure. In the same way, it is possible that a text that is suspected to have plagiarism can possess structural similarities with the text that is believed to be the source of the plagiarism. External plagiarism and parallel corpora detection algorithms can be used to find the similar text portions of the same literary work which is rewritten by different authors (such as the story of Leylâ and Mecnun). However, this hypothesis needs to be proven by an effective external plagiarism and parallel corpora detection algorithm and testing environment. By taking this goal as motivation, a novel external plagiarism and parallel corpora detection method is proposed in this study. The proposed method is further tested over PAN'09 external plagiarism dataset [2] and Leylâ and Mecnun literary works that are rewritten by different authors to observe if their writings' corresponding sections possess a significant similarity. Test results show that the proposed external plagiarism and parallel corpora detection method is able to detect similar texts successully.

Today, automatic plagiarism detection methods are accomplished in two different approaches. One of them is based on detecting the similarity of the source literary text with the texts that exist within the reference text dataset. This way of detecting the existing plagiarism is called external plagiarism detection. The second approach is based on detecting the plagiarism that exists in a suspicious text without having need a reference text dataset. Since this kind of approach does not need a reference text dataset, this approach of detecting the existing plagiarism is called intrinsic plagiarism detection. The plagiarism detection approach that is stated in this study is based on the first approach. It is based on the problem of detecting the plagiarized documents by making use of an existing reference text dataset. Since the proposed algorithm is dependent on a reference text dataset in the process of detecting the existing plagiarisms, the usage scope of the proposed external plagiarism detection algorithm could be further extended to analyze the similarity of the same literary texts that are rewritten by different authors, or the similarity between the texts that are the translations of the same source literary text in different languages. Hence the proposed algorithm is also evaluated if it is able to detect and quantify such similarities, namely, detecting parallel corpora.

1.2 Problem Statement

Textual plagiarism implies the usage of an author's text, his/her work or the idea that is inserted in another textual work without giving a reference or without taking the permission of the original text's author. Today, existing intrinsic and external automatic plagiarism detection methods tend to detect plagiarism cases within a given dataset in order to run these algorithms in a reasonable amount of time. Hence a reference document set is built in order to search for plagiarism cases successfully by these algorithms. Building a reference document set successfully is another scientific problem that needs to be solved. Some of the methodologies that are offered to build the reference document set can be listed as nearest duplicate search and nearest neighbor search. These methods can be used in detection of the documents that are partially or fully similar to

the particular suspicious document. In near duplicate detection methods, while detecting the documents that are similar to the particular suspicious document, a relational network is set up. Every node in the network represents a document while the edges between the nodes shows if the document pairs are related with each other. Relations between the nodes are inferred by an approach that is based on document fingerprints [10]. However in this study, since the purpose is to detect parallel corpora and external plagiarism, details such as formation of the reference document set is not further investigated.

1.3 External plagiarism and Parallel Corpora Detection

Analysis and detection of external plagiarism cases are highly related with the problem of detecting the texts possessing a similar structure. The aim of external plagiarism detection methods is to detect the existing plagiarism cases in a suspicious text document by making use of the reference text dataset and to observe the detected similar structure in the source texts that exist within the reference text dataset. Similarly, in the problem of detecting the parallel corpora, the structural similarity of a text that is particularly investigated with the texts that are rewritten by different authors in the same or in another language and based on that particular text is quantified. The main problem in parallel corpora detection is the automatic detection of the existing structural similarity of the texts and the quantifying this similarity. External plagiarism detection can be seen as a special form of the parallel corpora detection. In both of these concepts similar sections that exist in those particular documents are detected and quantified. Unlike parallel corpora detection, there is a preliminary step in which the documents that are suspected to be similar are detected.

The problem of detecting similar structures between different set of texts is not a new concept. One of the existing proposed methods is called “coupled clustering” [27]. Similar to the external plagiarism and parallel corpora detection method that is proposed in this study, the coupled clustering approach is based on

the principle "given two sets can be said to be similar as long as they possess a high number of common elements." As another independent research study, a similar study to compare and quantify the similarities between a literary text and its translations in other languages are carried out as Bilkent University Information Retrieval Group [11][1].

1.4 Research Contributions

In this thesis we

1. Propose a clustering-based similarity detection approach for analysis and evaluation of the similarity between the literary texts that have the same textual structure.

2. Show that our method, which is tested on various datasets such as PAN'09 plagiarism dataset and Leylâ and Mecnun works of different famous authors, provides competitive results with other methods.

1.5 Overview of the Thesis

The rest of the thesis is organized as follows. Chapter 2 provides extensive background information about the existing plagiarism detection and parallel corpora detection algorithms. Chapter 3 introduces our proposed plagiarism and parallel corpus detection algorithm. Chapter 4 describes our test collection which is PAN'09 plagiarism dataset and different versions of Leylâ and Mecnun that are rewritten or translated into other languages by different authors. Experimental results are reported in chapter 5. Finally, we conclude our work in the last chapter with a summary of our findings, future research pointers and last pages are reserved for informative data such as the definitions of symbols used and detailed evaluation results.

[1]The emphasized method will be further explained in the related work section.

Chapter 2

Related Work

The word plagiarism is derived from the Latin word *plagiarius* which literally means kidnapper to express stealing someone else's work. However its present use was introduced in English in the 17th century [1]. With the emergence of institutional and academical life and advance in technologies, the outcomes of plagiarism started to become more offending for the real owners of the plagiarized work. Today, plagiarism is a serious crime. In order to defend the rights of the original owner of the works, ways of detecting plagiarism are being investigated as a research field in computer science. Besides performing individual research studies in this field, every year PAN plagiarism workshop has been organized in order to improve or come up with new solutions to this problem since 2009. In the next two sections, background information about some existing external plagiarism detection and parallel corpora detection approaches are introduced.

2.1 External Plagiarism Detection (EPD)

In the case of external plagiarism detection, the particular suspicious text is compared with the source texts existing in the reference text dataset. The number of source texts may be in excessive amounts due to large size of reference text dataset. The number of comparisons between the suspicious text and source texts

may be beyond the acceptable time limits. In order to cope with that problem, the external plagiarism detection methods that are discussed in this section use a preliminary elimination within the source texts and leave a subset as candidate documents in the process of detecting the source of plagiarism.

As the second common feature, most of the discussed external plagiarism detection methods in this section have location of the plagiarized passages within the preselected candidate texts as a second step. These two steps are roughly enough for detecting the plagiarism. However, as it will be further discussed below, different algorithms usually have some other additional solutions to increase the accuracy of found plagiarism cases and these solutions form the rest of the steps for their algorithm. In the next sections, some of the external plagiarism detection (EPD) methods that are used in detecting the plagiarized fragment of suspicious texts are reported in detail.

2.1.1 Similarity Measure-Based EPD

Hariharan et al. [17] propose a plagiarism detection method for text documents. Their proposed method is composed of three steps. In the first step, documents are tokenized into sentences. In the second step, sentences go through preprocessing step. Stop words are eliminated and stemming is performed. Then in the third step, sentences are compared with each other by using cosine metric as measure [38] and the sentence pairs that have similarity above a certain threshold are considered as plagiarized. The method is tested over a corpus which was collected from a set of 120 students from different departments. Students are put under an exam like questions and they answered the questions and allowed to plagiarise. They also quoted the reference and exact passage where they did the plagiarism case for ground truth. Then the proposed method is compared with one of the avaliable commercial plagiarism detection tools, which also makes use of cosine similarity in plagiarism detection process, over the prepared corpus. According to experimental results the proposed method outperforms the commercial tool for the cosine similarity results.

2.1.2 Fingerprinting-Based EPD

A cluster-based external plagiarism detection method is proposed by Zou et al. [47] for PAN'10 competition. Their algorithm is based on fingerprinting technique and is composed of three steps: A preselecting step to narrow down the amount of source documents that will be compared with each suspicious documents. Second step is called locating which is the stage of detecting plagiarized sections between source and suspicious documents and this step makes use of fingerprints. Locating operation is a two stage approach and accomplished by clustering and merging of the fingerprints. Merging operation is done by using longest common subsequence algorithm and a proper threshold. Then by clustering step impact of obfuscated text on locating is reduced. Results of the proposed method over PAN'10 dataset shows that the proposed method is able to detect plagiarism with an overall score of 71%.

Kasprzak et al. [22] proposed a method for external plagiarism detection for PAN'09 competition which is already used as anti-plagiarism system in maintaining of the Czech National Archive of Graduate Theses. The method is composed of three steps. The first step is called tokenization where the words in Czech language are represented by using US-ASCII characters and some specified short words are not taken into consideration. In the second step, tokens are joined into chunks which are composed of four to six words. Then the chunks are hashed by hash function and indexed by using an inverted index. Hash values are mapped to the sequence of document IDs in that inverted index structure. In the third step they compute the similarities among the documents by making use of the previously created inverted index structure. The similarity of document pairs is calculated as the number of chunks in which the particular document pairs have in common. Working principle of the existing plagiarism detection method is only capable of which documents are plagiarized. However, PAN'09 are required to find the exact locations of the plagiarised texts. For this reason, in order to run this system over PAN'09 dataset they go over a list of modifications over the existing plagiarism detection method. Firstly they modified the tokenization step so that the documents also hold the position information of the words that

they contain. They also modified the inverted index structure to retain additional data. So that the modified method is capable of giving positions of the chunks for the list of documents that have the same hash values with the particular suspicious document. They also added a new postprocessing step to the existing method for the case of competition where they remove the overlapping passages for each suspicious document and keep only largest of them. Experimental results over the PAN'09 dataset show that the proposed method has good recall and overall values and they finished the competition in second place.

Schleimer et al. [40] proposed a method that is used in identification of similar portions of provided documents. Their method is called winnowing which is a local algorithm that makes use of fingerprinting technique. Their method first derives n-grams from the document, then by a hash function derived n-grams are hashed. Winnowing method just like other local algorithms, makes use of window concepts and selects one of the hashes as fingerprint within the bounds of that particular window. Winnowing algorithm selects the hash with minimum value from each window as fingerprint. The reason behind this approach is the belief that adjacent windows also tend to contain same hash value, thus making fingerprint of a document quite small with respect to its original size. The proposed method also uses two different thresholds to reduce the amount of noise and to increase accuracy. The conducted experiments over a large sized web data shows that the winnowing method successfully detects similar portions. Winnowing method is also adapted as a plagiarism detection software (MOSS) by one of the authors. They state that MOSS shows well performance without giving false positives and it is already being used professionally for years.

2.1.3 Indexing-Based EPD

Vania and Adriani [43] developed an external plagiarism detection method that is based on comparing passage similarities between the source and suspicious texts for the PAN'10 corpus. A passage is defined as a block with 20 sentences. Their approach is based on four steps: A preprocessing step to translate the texts into English in corpus which are in a language other than English. In their second

step they select a subset of source documents for each suspicious document as candidate plagiarism source documents. This is done providing suspicious texts as query to indexed source documents and retrieving the most similar top 10 source documents. The third step includes dividing top scoring 10 source and suspicious documents into passages then indexing and retrieving passages that have similar sections found in source documents. They only use top-5 similar source passages for each suspicious passage and remove other lowscoring passages which also forms their last step of method. According to evaluation results, their method performs around 90% with precision although their method is not as good as precision at recall and granularity criteria hence their overall plagiarism detection score is 13%.

Muhr et al. [30] propose an external plagiarism detection method for the PAN'10 competition that can perform plagiarism detection for translated and non-translated text documents. Their proposed algorithm is divided into two main steps. The first step is called retrieval step and in that step documents are divided into overlapping blocks and then indexed by using the Lucene indexing tool. They also store the information such as the offset and location of each block within the index. Similarly suspicious documents are also split into consecutive overlapping blocks and these blocks are treated as queries. By using these queries over the Lucene index retrieval operation is done to retrieve potentially plagiarized passages. Then, they apply some heuristics on potential matches to finalize the detection results such as limiting the window size and window step size. For non-English documents, they have an additional preprocessing step in which the documents are translated into English using word alignment algorithm. Word alignment algorithm translates a document into a specified language by trying to find the pairs of words what may be used as candidates of translation and already adopted by many translation systems. Their second step is called the postprocessing step and the potential plagiarized sections are filtered further in this step. According to this final step, a sequence of words in both texts is considered as a match if the sequence contains at least three consecutive words and has a length of at least 10 characters. According to experimental results over the PAN'10 dataset, the proposed plagiarism detection method shows a good

performance with an overall score of 69% and finishes the competition in the third place.

2.1.4 Longest Common Subsequence-Based EPD

Basile et al. [8] proposed an external plagiarism detection algorithm for the PAN'09 competition that is composed of three steps. In the first step, in order to reduce the number of comparisons that has to be done for each suspicious document, a subset of source documents are selected for candidate source documents of plagiarism containers. In this step, they make use of word length n-grams of both texts and calculate n-gram distance[1] of that particular suspicious document to each and every source document. By making such a preselection step, they significantly reduce the execution time of the algorithm with a recall of 81% which means a negligible amount of loss. After detecting the 10 plagiarism candidates for each suspicious document, in the second step algorithm aims to find the plagiarized passages. The goal in this second step is to detect the common subsequences between the source and suspicious documents that are longer than a fixed threshold. In order to accomplish this, they encode the original source and suspicious documents by T9 encoding. The idea of T9 encoding is to represent 3-4 characters with a single digit such as 2 represents the set {a,b,c} in this form of encoding. The texts are converted into T9 because authors claim that a long common subsequence in T9 form in almost unique and most probably denotes a plagiarism. In the final step, they check if the found common subsequences are in an order - following each other. In order to understand this they draw a plot and represent found sequences as dots. For the case of non-obfuscation plagiarisms, these dots turn into a line over the pilot whereas they turn into squares in obfuscated cases. By labeling these lines and squares as plagiarism cases they finalize the proposed algorithm. Experimental results over the PAN'09 show that the proposed method has good precision (67%) and recall (63%) performances and they finish the competition in the third place.

[1]N-gram distance and calculation details of the word n-grams will be further explained in proposed method section of this study.

2.1.5 Levenstein Distance-Based EPD

Scherbinin et al. [39] propose a method for detecting external plagarism by using Microsoft SQL Server platform. Their approach makes use of fingerprinting-based algorithm to compare the documents of dataset and Levenstein distance metric to detect the exact plagiarized passages within the detected particular documents. Their method is composed of four main steps. The first step is called preprocessing and at this step they use winnowing, which is one of the existing finger printing based algorithms and each document is replaced with a set of its hashes. The second step is called locating sources and at that step they reduce the number of documents before the process of plagiarized fragments detection. In this step, the pairs of documents which share at least one fingerprint are stored in a table for the next step. The third step is called detecting plagiarized passages and in this step common fragments within the documents of candidate set are detected by using Levenstein distance metric. In the final step they use Microsoft SQL Server Integration Services to export the plagiarism information in XML format adapt the results into competitions standards. According to results over PAN'09 dataset, the proposed approach has good results about precision and recall but the approach provides bad results about the granularity criterion hence they finish the competition as 6th.

2.1.6 N-Gram-Based EPD

Grozea et al. [16] propose and external plagiarism detection method that is based on plotting the plagiarism candidates with a method called ENCOPLOT and matching the pairwise sequences in linear time to detect the plagiarism case. Their proposed method includes two steps where in the first step they create a matrix of kernel values (a similarity value based on each source and each suspicious document) between each source and suspicious document. In the second step, each promising pair is further investigated to extract positions and lengths of the subtexts have been plagiarized by using ENCOPLOT method which is a scatter plot of a sublist of the positions where both texts have the same n-gram.

Positions are sorted by the value of the first index in each pair and from this list a contiguity score is derived. Then a Monte Carlo experiment is carried out to find the largest group pair and the plagiarism instance and extracted as output if all the tests are succeeded. By following this method they resulted in an overall F-measure score of around 79% with a low granularity of 1.0027.

Ferret is an external plagiarism detection tool and one of the competitors of PAN'09 that is proposed by Malcolm and Lane [26]. They define Ferret as a fast and interactive plagiarism detection tool which leaves the final decision if a found document pair contains plagiarism or not to the user of the tool. Ferret's working principle does not require direct comparison of the document pairs. According to Ferret algorithm, for every three word triplet of the suspicious document Ferret shows possible source of plagiarism documents to the user of the tool. Hence this way of analysing makes Ferret a very fast tool. However, authors state that several modifications to Ferret are needed to make it work with the PAN'09 dataset. The first problem they faced for the case of competition was the large scale of PAN dataset. In order to cope with that, they divided the competition dataset into batches so that in each batch file only a subset of suspicious documents were taken into account. The second problem was about the automation of the tool. Originally Ferret wasn't able to take final decision about a document if it contains plagiarism or not. So they needed to automate the decision if a document contains plagiarism. To cope with that problem they defined some thresholds such as number of consecutive detected triplets needed to label a document as plagiarized. According to experimental results over PAN dataset, Ferret performed a good recall 60% but quite low precision 3%. Authors defend that this is because of the unsuccessful automation modification of Ferret. And could have derived more successful results by adopting a different approach before giving the automated decision such as deriving a threshold for longest common subsequences of detected triplets.

2.1.7 NN-Search-Based EPD

Zechner et al. [31] proposes a plagiarism detection method for detecting both external and intrinsic cases by making use of vector space models. Their goal is to identify document passages that are partially derived from other documents where this derivation can be equal sequence, similar bag of words or similar phrases. Their proposed method is composed of three main sections. Firstly they apply to a preprocessing step where they identify the sentences of a given source document and cluster the sentences of the document. Later they store the sentence and assigned cluster pairs in an index structure. They do this process for all the documents of source text dataset. The second step is called retrieval step and in that step for a given suspicious document D_s they perform a preprocessing step just like they do previously for source documents and derive sentences and sentence clusters from the suspicious documents. Then they look up best matching source document clusters for the particular suspicious document sentence's assigned cluster. From the detected cluster pairs they take the k most similar sentences where the similarity is measured by cosine similarity measure. If a taken sentence pair is more similar to each other than a predefined threshold, they are labeled as plagiarized sentences. In the final of their proposed method, they do a merging operation of sentences provided that the sentences are occured consecutively. This final step is done to reduce the granularity of the detected plagiarism cases. Experimental results over a random sample of 500 suspicious documents from the PAN'09 corpus show that they detect the plagiarism cases with a precision of 60% and recall 37%.

2.1.8 Summary of External Plagiarism Detection Approaches

The external plagiarism detection approaches which are briefly discussed in the previous section are summarized in Table 2.1. Their overview of the strengths and weaknesses are also given in Table 2.2.

Table 2.1: Summary of external plagiarism detection approaches.

EPD Methods	Work	Blocking	Clustering	Preselection	Postprocessing	F Result
Similarity Measure Based	Hariharan Approach [17]	Yes	No	No	No	N/A[1]
Fingerprinting Based	Zou Approach [47]	No	Yes	Yes	Yes	0.75[3]
	Kasprzak Approach [22]	Yes	No	No	Yes	0.62[2]
	Schleimer Approach [40]	Yes	No	No	No	N/A[1]
Indexing Based	Vania Approach [43]	Yes	No	Yes	Yes	0.45[3]
	Muhr Approach [30]	Yes	No	No	Yes	0.77[3]
LCS Based	Basile Approach [8]	No	No	Yes	Yes	0.60[2]
Levenstein Distance Based	Scherbinin Approach [39]	No	No	Yes	No	0.62[2]
N-Gram Based	ENCOPLOT [16]	No	No	No	Yes	0.69[2]
	Ferret [26]	Yes	No	No	No	0.06[2]
NN-Search Based	Zechner Approach [31]	Yes	Yes	Yes	Yes	0.46[2]

1- No information is available , 2- Results from PAN'09 dataset, 3- Results from PAN'10 dataset

2.2 Parallel Corpora Detection

The process of detecting the parallel corpora can be accomplished by both supervised and unsupervised techniques. In this study, for detecting the parallel corpora we use unsupervised clustering techniques. Therefore, we only focus on similar parallel corpora detection methods that are based on unsupervised techniques in this section.

2.2.1 Coupled Clustering

Coupled clustering is a method for detecting structural correspondance between substructures of distinct textual writings [27]. The method aims to identify structurally similar subsets between the texts. Coupled clustering has an algorithm that is based on a cost function. The ideal cost function is detected experimentally. Later, in order to observe and evaluate the performance of the proposed method, they use an artificial dataset that is formed by holy books. The data coming from holy books are mixed and not initially classified in that dataset. Dataset is later clustered by using coupled clustering and in evaluation period, the results of the clustering operation are evaluted by scholar people of each religion that is included in that particular test. Later they compare the

sets that are found by scholar people for each religion and the results that are found by coupled clustering. As an outcome of this comparison, they calculate accuracy results for coupled clustering method. According to accuracy results, they observe that the findings of scholar people of each religion match with the clustering results of coupled clustering method significantly.

2.2.2 STRAND

STRAND is a method that is proposed for extracting the parallel (bilingual) text by mining the web data [36]. In the heart of this approach, the belief is that the translated web pages tend to show significant structural similarities with each other. Hence such structurally similar data can be labeled as parallel or similar by making use of an appropriate detection technique without having need to understand the content. The method is composed of three steps: The first step is finding the location of pages that may have parallel translations in which popular search engines are used in this detection process. The second step the identification of candidate pairs that may be translations. The pairs are generated automatically if there is only one candidate translated page for a source page. If the number of candidates are more than one for a specific site, then document lengths are taken into consideration because of the insight that the translations of one site to another site tend to be similar in length. In the final step, the elimination of nontranslation candidate pairs are done. They calculate a value called difference percentage from the web page pairs by making use of their html codes. Then by applying to Pearson correlation, they infer if this difference is significant or not. According to Pearson correlation results, the pages with significant difference are eliminated. The tests results that are made over the language pairs: English - French, English - Spanish, English - Basque and English - Arabic show that the STRAND is successful at extracting the parallel texts from the gathered web data.

2.2.3 Translation Relationship Index

Can et al. [11] proposed a method called Translation Relationship Index (TRI) for quantifying translation relationship between the source and target texts. Their method is language independent and in the process of quantification of these parallel texts, they make use of the texts' structural similarities. According to the method, they first partition the source and target cluster texts into blocks. Then, by making use of suffix trees they extract the base clusters from these blocks. The term base cluster stands for the nodes which are formed by word phrases that exist in suffix tree. Later target and source document blocks are clustered seperately. Translation relationship index is calculated from these formed clusters and it can be defined as source document's clusters' distribution average over the target document's cluster structure. The method is based on the hypothesis that similarities/dissimilarities among the source blocks are kept as similar and reappear in target blocks. Hence by using the method a significant similarity is expected between the source and target texts. For testing the proposed method, they use Shakespeare's sonnets and their translations in French, German, Latin, and Turkish. According to their experimental results, TRI method is successful in translation relationship quantification.

2.2.4 Fuzzy Set Information Retrieval

Koberstein et al. [24] proposed a method for detecting similar web documents by using word clusters. They propose a sentence-based fuzzy set information retrieval approach and use word clusters to capture the similarity between different documents. Compared documents do not have to be composed of the same words to be labeled as similar but the words that form the documents must be based on the same fuzzy-word sets. Words are included into fuzzy sets either partially or fuzzily and the words of a set possess strength of membership to that particular set. Three different fuzzy-word clustering techniques are proposed in the paper. The first technique is called correlation cluster. It only considers the co-occurences of words in documents to compute the word similarity. It uses the

occurence or absence of two words in each document and a correlation value is calculated. The second technique is called association cluster and it is constructed by considering the frequency of co-occurences. So this method takes into account how many documents a particular words pair occur together for at least a specified threshold times. The last technique is called metric cluster and it uses the distances among words in a set of documents as well as the frequency of occurences. Therefore, the words that occur together closer yield higher correlation values than the ones that occur far away from each other. Calculated correlation values by any of the three clusters are then used to compute the degrees of similarity of sentences in any two documents. The degree of similarity between any two documents is determined by the number of similar sentences in the documents. Experimental results over a large wikipedia web corpus show that the proposed detection approach has the best performance when metric clustering technique is used.

2.3 Paraphrase Extraction (PE)

If two text fragments carries the same statement with different expressions, these corresponding text fragments are said to be paraphrases of each other [29]. The following paraphrase extraction (PE) approaches are used to detect paraphrasing within bilingual texts. They are tested in available parallel corpora datasets. These algorithms are used to detect similar texts in parallel corpora and in that sense they possess a similar aim with the proposed parallel corpora and external plagiarism detection approach in this study. Hence these approaches are discussed in this section.

2.3.1 Unsupervised PE

Barzilay et al. [7] proposed an unsupervised learning algorithm for detection and extraction of paraphrases from a corpus which contains different English translations of the some famous classic novels. During the prepocessing step of their

method, a sentence alignment operation is performed. They believe that the sentences which are translations of the same source text tend to contain a high number of identical words which can later be used in the sentence matching operation. After the preprocessing step, they make use of a part-of-speech tagger to label the noun and verb phrases in sentences. Detected identical words are used to learn context rules and in application of these rules. Then the proposed method finds the similarity of sentences in their local context. If the contexts surrounding the two suspected phrases are similar enough, then the suspected phrases are labeled as paraphrases. In order to understand if the contexts are similar, they propose a three-step co-training algorithm. The first step is called initialization and the words appear in both sentences of aligned pairs are used to create initial seed rules. In that step, they make use of identical words that appear in both sentences. But this approach does not necessarily give successful results. In the second step, they use contextual classifier which uses the previously detected initial seeds and train the classifier with the contexts around positive and negative paraphrasing examples. Which of the available contexts are strong predictors for paraphrasing is found at this step by comparing the contexts positive and negative paraphrasing counts. In the final step, context rules that are extracted in the previous step are applied to the corpus to derive a new set of positive and negative paraphrasing examples. The results of the proposed paraphrase extraction method is evaluated in terms of accuracy and recall, human evaluation and Kappa (κ) measure. According to the results, the proposed algorithm provides high amount of correctly paraphrased sentences from their parallel corpus dataset.

Bannard et al. [6] proposed a paraphrase detection and extraction method that is similar to Barzilay et. al's approach. The main difference between these two approaches is that Bannard et al.'s method is capable of working in bilingual environment and they use a bilingual parallel corpora to evaluate their paraphrase extraction algorithm. They define a paraphrase probability that allows paraphrases to be extracted from bilingual parallel corpus to be ranked using translation probabilities. The essence of their method is to align the extracted paraphrases from bilingual corpus and equate different English phrases that are

aligned with the same phrase in another language. Their proposed method is composed of two main parts: In the first part, they are either automatically or manually extract paraphrases. Since they work in a bilingual corpus and contexts around the paraphrases are highly tentative with respect to monolingual corpus, unlike the work of Barzilay et al., they do not consider the contexts for identifying the paraphrases. Instead they use phrases from other languages as pivots and look at how certain phrases are translated into another language from English. Unlike the method of Bannard et al., they extract more than one possible paraphrase for each phrase and then assign a possibility to each of the possible paraphrases. The probability of phrases is actually a conditional probability and be calculated as using maximum likelihood estimation by counting down how often the original phrase and its translated version are aligned in the parallel corpus. They test their method in a large German-English bilingual corpus. According to the results, when they make the alignment manually, their method is able to detect and extract the paraphrases more accurately. They also found that when they perform a word sense disambigation for the cases where they make the alignment automatically they observe that the automatical way of extracting and aligning the phrases give closer results to the manual case.

2.3.2 Using TF.IDF scores for PE

Bengi Mizrahi [29] did a study on paraphrase extraction from parallel news corpora. Goal of the study is to create a database of paraphrases for generic use and in order to accomplish that a method for the extraction of paraphrases from news articles regarding to same event is proposed. The approach is composed of three steps. In the first step, news articles pairs that carry the information about the same events are collected. Then news corpus is indexed, matched with each other and the matches are ranked according to TF×IDF scores. Finally, highest ranked documents are picked from the corpus. Second step is called sentence-level matching in which equivalent sentence pairs are collected out of the news article pairs. In order to accomplish that some of the existing machine translation methods such as BLEU-N, WER, PER, and NIST-N (for comparison) are

used and applied to their documents matches. Third step is called phrase-level paraphrase extraction and in this final step the paraphrases are extracted from sentence pairs. First sentences are parsed and dependency trees are obtained. Later common nouns are searched between the nouns in each tree and paired. Finally the path with highest frequency count of internal relations is returned from the dependency tree. According to evaluation results, n-gram based sentence level matching approaches catch the sentences with giving less false positives for paraphrase extraction and the proposed system extracts the paraphrases with an accuracy of 66%.

Table 2.2: Overview of external plagiarism detection approaches.

EPD Methods	**Work**	**Advantages**	**Disadvantages**
Similarity Measure Based	Hariharan Approach [17]	(a) Better performance than some commercial tools (b) Simple implementation	(a) Tested over small corpus
Fingerprinting Based	Zou Approach [47]	(a) Fast execution	(a) Only for monolingual plagiarism detection (b) Cannot handle highly obfuscated text
	Kasprzak Approach [22]	(a) Already being used by Czech Government	(a) Only for monolingual plagiarism detection (b) Cannot handle highly obfuscated text
	Schleimer Approach [40]	(a) Already being used as a professional plagiarism detection tool (MOSS)	(a) Non-robust winnowing has weakness if the compared strings have low entropy
Indexing Based	Vania Approach [43]	(a) Precise results over bilingual corpus	(a) Fails to detect modified plagiarisms (b) Bad recall results over bilingual corpus
	Muhr Approach [30]	(a) Bilingual plagiarism detection (b) Intrinsic plagiarism detection support	(a) Poor intrinsic plagiarism detection performance
LCS Based	Basile Approach [8]	(a) Good F results over monolingual corpus	(a) Has many parameters to be tuned over specific corpus
Levenstein Distance Based	Scherbinin Approach [39]	(a) Precise results over monolingual corpus	(a) Depends on third party software
N-Gram Based	ENCOPLOT [16]	(a) Good F results over monolingual corpus (b) Low granularity scores	(a) Slow execution time compared to indexing based methods
	Ferret [26]	(a) Fast execution (b) Easy to use interface (c) Good recall results over monolingual corpus	(a) Poor precision results over monolingual corpus (b) No real support for finding plagiarized passages
NN-Search Based	Zechner Approach [31]	(a) Fast execution (b) Good precision results over monolingual corpus	(a) Too many corpus specific parameters to be tuned

Chapter 3

Plagiarism and Parallel Corpora Detection Method: P^2CD

The plagiarism and parallel corpora detection method (P^2CD) that is used in this study requires a specified corpus from which the possible cases of plagiarism and parallel corpora are detected. P^2CD compares the documents within the corpus with each other and according to their structural content similarity, plagiarism cases are identified. Each document in the corpus are represented using the vector space model [37] and by making use of the structural similarities of document index clusters, similarities of the documents are identified and quantified. In the rest of this section, the working principle of P^2CD is further explained in detail.

3.1 Components of the Method

3.1.1 Preprocessing

This step has the purpose of cleaning source and target document texts from punctuation marks, and from other symbols that are irrelevant with the document's context. In addition, elimination of the frequent words that are used within the language of document texts are also accomplished at this step. In

order to accomplish this, we adopted well prepared stopword lists. One is in English and constructed for the SMART information retrieval system at Cornell University, composed of 571 words. The other one is in Turkish [21] which is the manually extended version of other existing Turkish stopword lists that are used in study [12].

3.1.2 Blocking

This is the step where source and suspicious documents are divided into blocks. Relevant documents are divided into consecutive and sequential text portions called blocks. Sometimes size of texts would be not enough for dividing the text into many blocks and in such cases blocking can occur in a natural way. In such cases texts itself is considered as one block. In this phase the notion of sliding window size is also taken into account and the blocks are created in sequence from the place where sliding window size ends beginning from the start of previous block (see Figure 3.1). Sliding window concept is already used in studies like [3], [4] and [23] and its usage is shown to be effective for similarity detection problem. Size of the created blocks are kept the same in terms of the amount of words they contain. In the experiments the best performing block size is obtained by trying different sizes. The size of blocks should be equal both in suspicious and source documents (it will be discussed in the next section) because the blocks should contain the same amount of documents for P²CD to work properly. However, it does not mean that the obtained block number should be same both for suspicious and source documents.

3.1.3 Creating Documents

The blocks which are created from suspicious and source textual documents are divided into smaller textual parts in this step. These smaller parts are called documents and they are the smallest textual units in P²CD. Similar to previous step, relevant blocks are divided into consecutive and sequential documents.

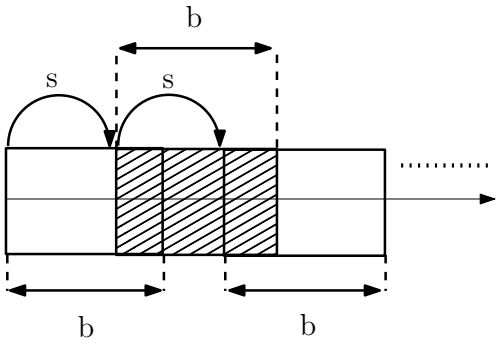

Figure 3.1: Sliding window-based blocking: l: total text length, b: block size ($0 < b \leq l$), s: step size, nb: number blocks, $nb = 1 + \lceil \frac{(l-b)}{s} \rceil$ for $0 < s \leq b$, $nb = 1 + \lfloor \frac{(l-b)}{s} \rfloor$ for $s > b$ adapted from [20].

Unlike blocking step, there is no sliding window concept while creating the documents. Size (in terms of words) and the number of created documents are set to be fixed both in source and suspicious blocks and in the experiments these attributes are tried to be optimized heuristically.

As an example for steps blocking and creating documents, 2 blocks that are derived from a relatively short suspicious text that exists in PAN'09 dataset are given below in Table 3.1.

Table 3.1: A passage from suspicious text 12648 and its first 2 consecutive blocks when blocksize: 25 words, step size: 10 words, and document size: 5 words.

Suspicious text 12648			
The poetical as well as moral decline of taste in our time has been attended with this consequence, that the most popular writers for the stage, regardless of the opinion of good judges, and of true repute, seek only for momentary applause; while others, who have both higher aims, keep both the former in view, cannot prevail on themselves to comply with the demands of the multitude, and when they do compose dramatically, have no regard to the stage.			
First Block		Second Block	
Doc. No. 1	The poetical as well as	Doc. No. 1	our time has been attended
Doc. No. 2	moral decline of taste in	Doc. No. 2	with this consequence that the
Doc. No. 3	our time has been attended	Doc. No. 3	most popular writers for the
Doc. No. 4	with this consequence that the	Doc. No. 4	stage regardless of the opinion
Doc. No. 5	most popular writers for the	Doc. No. 5	of good judges and of

3.1.4 Indexing

In this step, the documents which are created from the blocks of source and suspicious textual documents in the previous step, are utilized. The words from these corresponding documents are taken into account and they are used as document description vectors using the vector space model [37]). Since the number of documents within each block are fixed, the size of indices created from the blocks after indexing step are also equal for suspicious and source texts and this lead to a healthy comparison of the both texts by making use of their term vertices.

3.1.5 Block Matching

The documents which are derived from suspicious and source documents are indexed in the previous step. In this step, these indexing structures are compared with each other in order to understand if a particular suspicious text block resembles any of the existing source text blocks. Hence for each suspicious and source block there exist an indexing structure. In the case of texts which are composed of many blocks, a huge amount block comparisons are needed which is infeasible to accomplish. In order to diminish this workload with a reasonable amount of loss, in this step the blocks which resemble each other are preselected by considering only their indexing structures without applying any other costly operations such as clustering. In the process of comparing index structures, cover coefficient clustering method (C3M) is utilized in order to find the number of clusters that will be created from the index structures. According to the cover coefficient concept, the number of clusters can be estimated from an index structure by using the formula [13]:

$$n_c = \frac{m \times n}{t} \tag{3.1}$$

In the above formula n_c denotes the number of clusters, m stands for the number of rows exists in the document-term matrix of indexing structure, similarly, n

stands for the number of columns in the document-term matrix of indexing structure, and t is the total number of non-zero elements in the indexing structure. The blocks, which have the similar (the ones that are close to each other than a certain threshold) number of clusters that is found by using the above formula according to their indexing structures, are considered for more detailed analysis.

3.1.6 Clustering

The qualified blocks of source and suspicious blocks, which have similar number of clusters, go through a clustering operation in this step. Clustering algorithms are designed to differ the elements of a given dataset so that the similar elements are put into the same cluster while elements of the different clusters show significant difference from each other. In order to cluster the qualified blocks, cover coefficient clustering algorithm (C3M) is used. Note that block documents of suspicious and source blocks are clustered seperately. By making use of the previously constructed block index structures and the cluster numbers, C3M clusters the documents of the blocks so that the finals clusters are derived that will be used in the next step.

3.1.7 Comparison of Cluster Distributions

In this step, cluster formations (clusterings) that are derived by clustering the documents of source and target text blocks seperately in the previous step are compared in order to measure the similarity between them. If we call the clusters that are derived after the clustering operation that is accomplished in the previous step as C_s for source text clustering formation and C_t for target (suspicious) text clustering formation, in order to infer that the corresponding blocks of these cluster formations have a structural similarity of a case of external plagiarism, there should be a meaningful similarity (eg. the similarity should be significantly different from random case) between the cluster formation C_s and C_t [11]. For this reason in this step, the distribution of the elements of C_s over C_t is calculated.

For example, if we consider the block that will be clustered has the list of documents {a, b, c, d, e, f, g}, then the target text clustering formation C_t may follow a cluster distribution such as {a, b}, {c, d}, {e, f}, {g}. Similarly, the source text clustering formation C_s may follow a cluster distribution such as {a', b', c'}, {d', e'}, {f', g'}. In this given example, documents a and a' represent the corresponding documents in target and source cluster formations. In order to find the structural similarity between the cluster formations C_t and C_s, we need to find the cluster elements' distribution of cluster formation C_s over the cluster formation C_t. Elements of cluster C1 is not distributed and all of them go to cluster C1', For C2 number of distributed clusters is 2, for C3 it is 2 and for C4 it is 1 (Detailed explanation can be observed in Figure 3.2). In other words, cluster distribution average of the documents that are found in the cluster formation C_s over the cluster formation C_t is(1+2+2+1)/4=1.5. Note that if we change the direction of the calculation and calculate the result as cluster distribution average of the documents that are found in the cluster formation C_t over the cluster formation C_s, the result does not have to be the same with the previously found result. For example, in this example when we change the direction of the calculation we find the result 2.0. In the ideal case of match between the cluster formations, the cluster distribution average of the documents is expected to give 1.0. A perfect match between the cluster formations of the different texts is rather unusual. For this reason, in the case of such perfect matches between the C_s and C_t clustering formations, it is suspected that the relevant blocks of the source and target texts possess similar structure or a potential plagiarism case (In this approach the number of documents that the clusters have is neglected but the cluster distribution average of cluster formation C_s over the cluster formation C_t is calculated as it is stated below and in the evaluation phase since both distributions are taken into consideration, the number of elements that the clusters have does not constitute a problem).

In order to calculate how the distribution of the elements of C_s cluster formation over C_t cluster formation can be in random case, a measure which is found by Yao [46] and mainly used in the field of databases is adopted. Originally Yao's

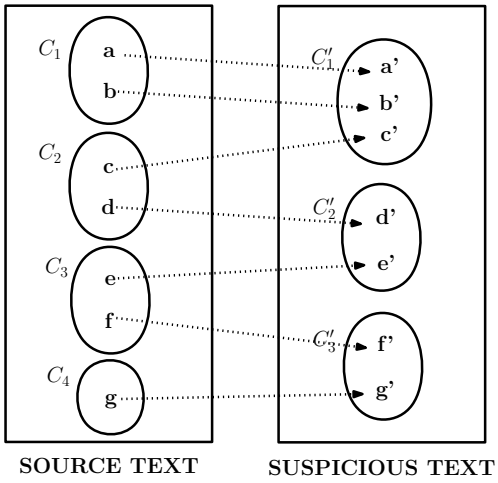

Figure 3.2: Distribution of source text cluster members into suspicious (target) text clusters.

formula determines the number of disk pages to be accessed to retrieve the related records of a query under the assumption that database records are randomly distributed among fixed size pages. Later, Can and Ozkarahan [13] adapted the formula for environments with different page (cluster) sizes. To use Yao's formula for the problem of external plagiarism and parallel corpora detection, we can treat the individual clusters of C_s as queries and determine how their members are distributed in the clusters of C_t. According to Yao, the number of target clusters for individual queries (individual clusters of C_s) in the case of random clustering (of C_t) can be obtained by using the following formula.

$$\begin{aligned} n_{tr} &= P_1 + P_2 + ... + P_{n_c} \\ P_j &= [1 - \prod_{i=1}^{k} \frac{m_j - i + 1}{m - i + 1}] \\ &where\ m_j = m - |C_j| \end{aligned} \tag{3.2}$$

Here we assume that we have m number of documents and n_c number of clusters in C_t and each cluster of C_t have a size of $|C_j|$ for $1 \leq j \leq n_c$. Now we consider the individual documents of C_s one by one. Assume that the cluster we consider from C_s, C_{source}, contains k number documents. In C_t the probability

that cluster C_j is selected is shown by P_j. Then the total number of target clusters to be accessed for the cluster C_{source} of C_s is equal to the summation (P_1 + P_2 + . . . + P_{n_c}). Average number of target clusters to be accessed for all clusters of C_t, is straightforward [46, 13]. During the randomization, the number of clusters and the size of the individual clusters of C_t are kept the same as they are obtained from the target text. However, it is assumed that the target text blocks are randomly distributed in C_t. These cluster distribution averages for random cases will later be used in the decision making step of P²CD.

3.1.8 Decision Making

Main goal of this step is to conclude if the corresponding suspicious and source text blocks have a significant similarity or not. The actual distribution value which is found by calculating the cluster distribution average of the cluster formation C_s into cluster formation C_t is compared with the distribution value of the elements of cluster formation C_s into cluster formation C_t in random case (random distribution value) which is found by the Yao's formula. If the actual distribution value between the cluster formations C_s and C_t is greater than or equal to the random distribution value, then it is inferred that the compared block pair does not have a structural similarity or a plagiarism case. If actual distribution value is less than the random distribution value, then it is looked if the found actual distribution value between the compared block pair is significantly different from random distribution value. If it is concluded that the actual distribution value between the cluster formations C_s and C_t are "significantly" different and less than the random distribution value then the compared blocks are labeled as the possible holders of a structural similarity or a plagiarism case.

3.1.9 Postprocessing

This final step is to conclude if the blocks that are found by the previous step carry a real plagiarism case. This step is also used to merge the found blocks which carry a plagiarism case but the plagiarism is observed within more than

one block. Hence, if a plagiarism allocates more than one block of suspicious text, after this step the found plagiarism can be better identified within a single and bigger text fragment. Thus lowering the granularity of the found plagiarism case[1]. In order to understand if the blocks that are labeled in the previous step carry a real plagiarism case two parameters are taken into consideration in this step.

1. A parameter called gap threshold to determine the size of gap between the labeled consecutive blocks. If the labeled block numbers are closer to each other than the gap threshold, then these blocks are deduced as plagiarized blocks. For example let's say the gap threshold is set to 2 and the block numbers {5, 6, 8, 9, 13, 20} of the suspicious text are labeled as a possible case of plagiarism. Then, according to gap threshold blocks 5, 6, 7, 8 and 9 are concluded as plagiarized blocks whereas block numbers 13 and 20 are excluded since these blocks have a distance to each other and to the other blocks that is larger than the gap threshold.

2. A parameter called consecutiveness threshold to check how many of the labeled blocks should be in sequence in order to consider those blocks as a possible case of plagiarism. For example let's say the consecutiveness threshold is 3 and the block numbers {5, 6} and {10, 11, 12} of the suspicious text are labeled as possible cases of plagiarism. Then, according to consecutiveness threshold blocks 10, 11 and 12 are concluded as plagiarized blocks whereas block numbers 5 and 6 are excluded since consecutiveness of 2 blocks is not enough when the consecutiveness threshold is set to 3. A similar parameter is also proposed by Needleman at al. [32] to detect the similar amino acid sequences between the compared proteins. The method they propose considers if the detected similar amino acids are in sequence and if they are in sequence, it accepts the longest similar aminoacid sequence as the similarity result.

As it can also be observed from the above parameters, labeled blocks are accepted as plagiarized if these labeled blocks follow a sequence among the labeled

[1]The term granularity will be explained in detail in experimental results section.

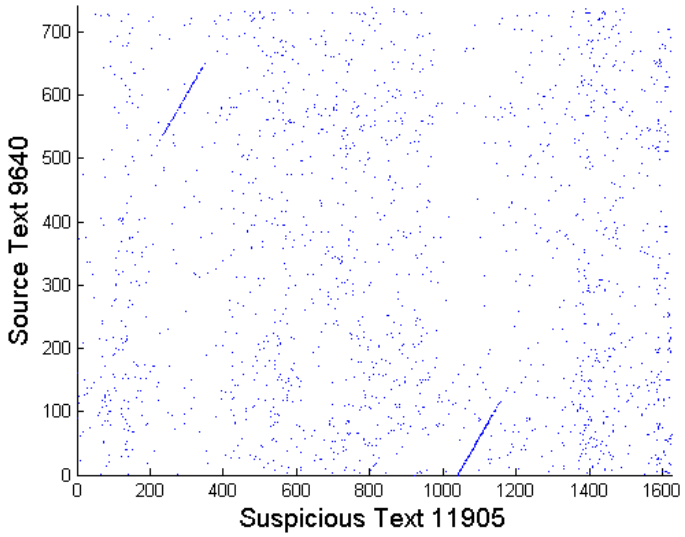

Figure 3.3: Scatter plot of detected blocks by P²CD for suspicious text 11905 vs. source text 9640 from PAN'09 dataset.

blocks. This is due to the belief that the found plagiarism will usually allocate more than one blocks. Hence the found plagiarism will be divided into consecutive blocks in this case. For this reason, P²CD is trying to detect such sequences. And in case of a found sequence, they are accepted as a case of plagiarism by considering the above thresholds. An example to visualize the case is given in Figure 3.3. In this figure, each dot represents the detected blocks that enters to postprocessing step and the actual plagiarism cases are clearly visible as lines as a combination of consecutive dots. By making use of gap threshold and consecutiveness threshold, P²CD's postprocessing step eliminates all the noisy dots and successfully detects the actual plagiarism cases.

3.2 Pseudocode of P²CD

Pseudo algorithm of P²CD is given in Algorithm 1 below.

Algorithm 1 P²CD Algorithm

```
1: Do preprocessing over suspicious and source texts.
2: Divide suspicious and source texts into equal sized and equal window-sized blocks (In this
   way we obtain b1 suspicious text blocks and b2 source text blocks. b1 does not have to be
   equal to b2).
3: Divide suspicious and source blocks into equal sized documents (Since block sizes are equal
   in suspicious and source texts, number of documents per block will be equal for both texts).
4: Create document by term matrices from the documents that are derived from suspicious
   and source texts.
5: Calculate the n_c cluster number for each block of both texts.
6: for Suspicious Block No = 1 to b1 do
7:     for Source Block No = 1 to b2 do
8:         if Selected blocks have similar n_c then
9:             Cluster the selected blocks seperately.
10:            Calculate the cluster distribution average between the selected
11:            source and suspicious blocks (actual distribution).
12:            Calculate the cluster distribution average between the selected
13:            source and suspicious blocks with Yao's formula (random distribution).
14:            if actual distribution < random distribution then
15:                Do postprocessing over selected suspicious and source blocks.
16:            end if
17:        end if
18:    end for
19: end for
```

3.3 Illustration of P²CD

Working principles of P²CD are illustrated in Figure 3.4.

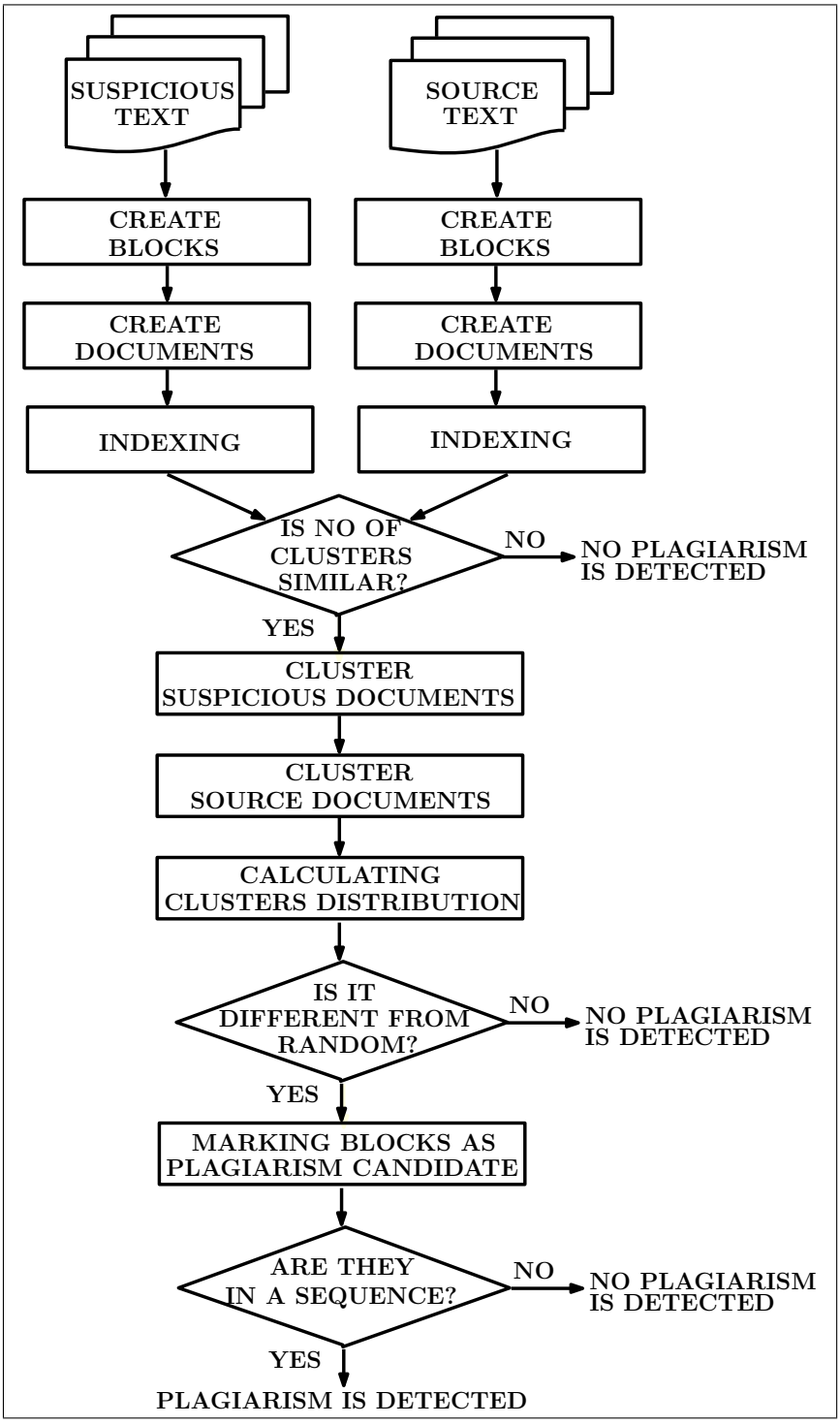

Figure 3.4: Illustration of P²CD.

Chapter 4

Experimental Environment

In this section, the experimental setup of this study will be explained. The experiments are performed over three different textual sources. First one is PAN'09 dataset over which we performed our external plagiarism detection tests. Others are Bilkent information retrieval group near duplicate dataset and several versions of the famous literature piece Leylâ and Mecnun written by different authors. These two main textual sources are used to detect the structurally similar parts within the texts' themselves. Hence by using them we tested if P^2CD is successful as a method of detecting parallel corpora.

4.1 PAN 2009 Plagiarism Dataset

PAN 2009 dataset is an experimental dataset that is prepared for the PAN 2009 international plagiarism detection competition for testing the methods of the different competitors over the same dataset for achieving comparable results. In the competition intrinsic and external plagiarism methods are evaluated seperately. Hence the dataset is composed of two main parts. One part is prepared intentionally for testing external plagiarism detection methods, and the other one is designed for testing intrinsic plagiarism detection methods. These parts are composed of source and suspicious documents which may contain plagiarism cases

that are artificially created by the organisers of the competition. In addition, every suspicious document has an accompanied XML file which contains the information of plagiarism cases such as from which source document the plagiarism is done, from which passage the plagiarism starts within that document, and the length of plagiarism. These information that are provided within the XML file for every suspicious document is accepted as the ground truth and they are used in the evaluation phase of P^{2}CD. As Mauer et al. [28] states, the the difference between a hard-effort original work and a plagiarized work can be murky. Therefore, one can say that the plagiarized cases also have levels for detection difficulty. In PAN 2009 dataset, this detail is not neglected and the accompained XML files also contain a confusion level. The plagiarism cases are done with an operation called 'obfuscation synthesis' [35] so that directors of the competition simulated the behaviour of a plagiarist who can modify or rewrite the sections they take from the original document. In this process, they made use of three different techniques. First one is random text operation so that the plagiarisms are done by randomly replacing the phrases from the source documents. The second one uses semantic vartions (e.g. synonyms, antonyms, hyponyms) of the source text while injecting them to the suspicious text. The third is based on shuffling the words of the plagiarized passage but maintaining the original part-of-speech sequence. As it can also be understood from the working principle of the obfuscation sysnthesis, artificially created obfuscated plagiarism cases do not necessarily be meaningful. The suspicious documents of the plagiarism dataset that is used in experiments contains different kinds of plagiarism cases. These can be listed as the documents with raw plagiarism (without obfuscation), documents with low obfuscation and the documents with high obfuscation. In this study the actual experiments is conducted over a large subset of the PAN'09 dataset. P^{2}CD is ran over 300 randomly selected suspiciuos documents which contains 100 non-obfuscated (raw) plagiarism cases, 100 low obfuscated plagiarism cases and 100 high obfuscated plagiarism cases. We were not able to consider all suspicious documents that exist in the dataset because of the running cost of P^{2}CD. Actually, running time of the method could be greatly reduced by considering very large block size and step size values. However, this time the method would quite likely fail to detect the existing plagiarism cases. Therefore, we were faced with a trade off between

the accuracy and the size of experimental dataset. At the end we decided to reduce to experimental dataset into 300 suspicious documents and conduct our experiments. Testing over subset of a PAN'09 dataset is also done by some of the competitors such as [31]. In their study, they reported their test results for a sample of 500 suspicious documents.

We provide information such as average number of words (μ) per suspicious document and their standard deviation (σ) for such averages integer for our PAN'09 sample dataset in Table 4.1 and for the whole PAN'09 dataset in Table 4.2. While giving the statistics, values are rounded to integer whenever it is needed.

Table 4.1: Statistics about PAN 2009 sample plagiarism dataset we use in the experiments.

No. of source documents	3000
No. of suspicious documents	300
No. of artificial plagiarism cases	1235
Avg. no. of plagiarism per suspicious document	5
Source document with max. length	444567
Source document with min. length	386
μ per source document	41172
σ per source document	56513
Suspicious document with max. length	352586
Suspicious document with min. length	1076
μ per suspicious document	27662
σ per suspicious document	50780

Below we also provide the distribution of plagiarized passage character lengths in our PAN'09 sample dataset in Figure 4.1 and in the whole PAN'09 dataset in Figure 4.2. There are no plagiarized passages in both datasets that has character length that is greater than 30000. However, as it can be noticed the lengths of the plagiarism cases follow a strange distribution and there is no plagiarized passages with character length between 6500 and 12000.

In addition to steps of the method which are expained in detail in the previous chapter, there are some other steps which are also used in the case of detecting plagiarism within the PAN 2009 plagiarism dataset. These steps can be called

Table 4.2: Statistics about the whole PAN 2009 plagiarism dataset.

No. of source documents	14429
No. of suspicious documents	14428
No. of artificial plagiarism cases	73522
No. of suspicious documents without plagiarism	7214
Avg. no. of plagiarism per suspicious document	11
Source document with max. length	578024
Source document with min. length	8
μ per source document	28698
σ per source document	49938
Suspicious document with max. length	456721
Suspicious document with min. length	328
μ per suspicious document	38031
σ per suspicious document	47820

"candidate source documents detecting problem" and "extracting the real plagiarim cases".

4.1.1 Candidate Source Documents Detection Problem

Since PAN'09 dataset contains thousands of suspicious and source documents, P^{2}CD has to make millions of comparisons in order to detect the existing plagiarism cases. However, this sounds quite impractical and can take months to process all comparisons. For this reason in the phase of detecting the case of plagiarism for every suspicious document that exists in the dataset, we are forced to reduce the amount of source documents to be compared with the respective suspicious document. Hence, each suspicious document is compared with the most similar 10 source documents that are previously detected. A similar approach was previously applied by Basile et al. [8] over authorship detection problem and over the PAN'09 competition and they achieved successful results with high recall. In order to find the most similar source documents to a particular suspicious document, firstly the distance between the particular suspicious document and every source document that exists in the dataset is calculated. Then, for each suspicious document, the closest 10 source documents are selected as the source documents that are candidate source of plagiarism and provided to P^{2}CD.

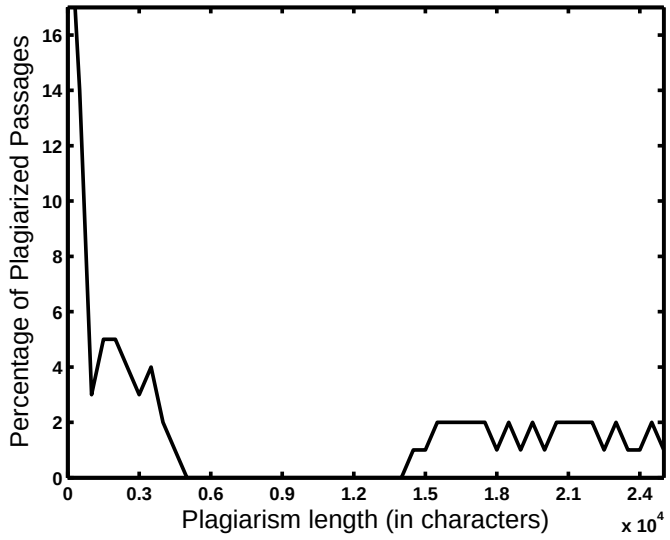

Figure 4.1: Distribution of plagiarized passage lengths in PAN 2009 sample plagiarism dataset we use in experiments.

In order to finish the calculation of distances between the particular suspicious document and source documents in a reasonable amount of time, all documents are converted into sequences of word lengths. For example, the sentence "To be, or not to be" is represented as 222322. The words which have character lengths greater than 9 is accepted represented by 9. By making this conversion, document lengths are greatly reduced and this helps to reduce the cost of detecting the candidate source documents. The distance between the particular suspicious document and source documents is calculated as comparison of the 8-gram distance frequencies. An example n-gram distance calculation with 8-grams is given in Figure 4.3. The distance between given two texts equals to 1 (worst case) since there is no common 8-grams between the provided texts. When we replace the first 8 words of the first sentence with the first 8 words of the second sentence, both sentences become identical and this time n-gram distance formula gives 0 as distance (best case).

8-gram distance is chosen in the tests because it is proven that this particular value gives experimentally successful results over the similar problems [9]. For the

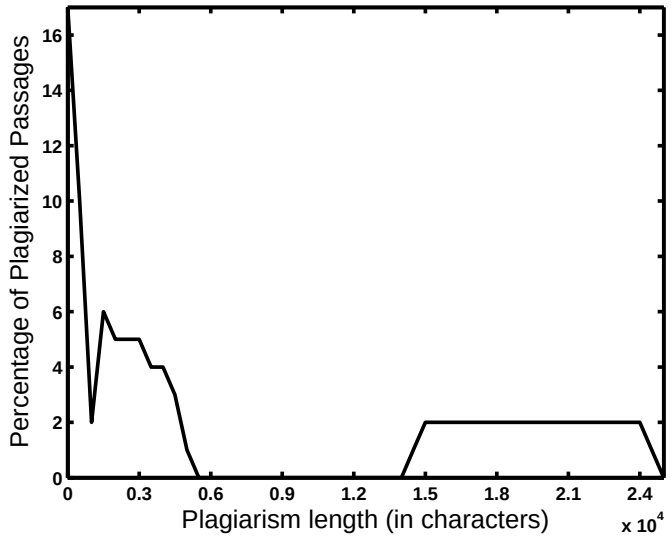

Figure 4.2: Distribution of plagiarized passage lengths in the whole PAN'09 dataset.

8-grams which exist either in document x or in document y and have a frequency that is greater than 0, the distance between the documents is calculated as:

$$d_n(x,y) = \frac{1}{|D_n(x)| + |D_n(y)|} \sum_{\omega \epsilon D_n(x) \cup D_n(y)} \left(\frac{f_y(\omega) - f_x(\omega)}{f_y(\omega) + f_x(\omega)} \right)^2 \tag{4.1}$$

Here ω denotes the respective n-gram, $f_x(\omega)$ denotes the frequency of n-gram ω in document x and $D_n(x)$ stands for the set of n-grams that exist in m document x has frequency greater than 0. Similarly $f_y(\omega)$ denotes the frequency of n-gram ω in document y and $D_n(y)$ stands for the set of n-grams that exist in m document y has frequency greater than 0.

4.1.2 Extracting the Real Plagiarism Cases

After executing P^{2}CD, in order to understand if the detected source and suspicious blocks really do contain case of plagiarisms, the XML documents that

Text 1: The term information retrieval (IR) in this paper is equivalent to document retrieval.

Text 1 in form of word lengths: 3499224529289

8-grams: 34992245, 49922452, 99224529, 92245292, 22452928, 24529289

Text 2: In this paper, the term information retrieval (IR) is equivalent to document retrieval.

Text 2 in form of word lengths: 2453499229289

8-grams: 24534992, 45349922, 53499229, 34992292, 49922928, 99229289

$d_n(x,y) = \frac{1}{(6+6)} \times (\frac{1}{1} + \frac{1}{1} + \ldots + \frac{1}{1}) \Longrightarrow d_n(x,y) = \frac{1}{12} \times \frac{12}{1} = 1$

Figure 4.3: N-gram distance calculation between given two texts.

are provided for every suspicious documents are used. These XML documents contain details about the corresponding source documents from which the plagiarism cases are done such as the starting offset and the length of plagiarism. By that way, the plagiarized passages are exracted from the source documents and the suspicious document. Then the blocks that are detected as plagiarized are compared with the ground truth plagiarized blocks that are derived by using the XML file and the results are used in the evaluation phase of P^{2}CD.

4.2 Leylâ and Mecnun Parallel Corpora Dataset

"Leylâ and Mecnun" which is originally written by Nizamî in Persian (born on 1141, died circa 1209) and has a great importance for Turkish and middle eastern literatures, is rewritten by many different authors as the time passed in history. Besides having different editions of the same piece by different authors, one can easily find the same literary work translated in another language. We believe that these different editions and translations of Leylâ and Mecnun writings should possess a significant similarity since they are all originally based on the same literary work. One of the contributions that will be achieved by this study is using P^{2}CD over these different editions and translations of Leylâ and Mecnun

literary work to understand if they are structurally similar to each other.

In order to achieve these goals and use in experiments, translation of the original Fuzûlî's (born on 1483, died circa 1556) Leylâ and Mecnun in Turkish latin script [15], in English latin script [18], rewritten edition of Leylâ and Mecnun by Nizamî [42] in Turkish in prose form are obtained. All Leylâ and Mecnun scripts are composed of short chapters. However, chapters of rewritten editions does not need to be similar with the original text. Hence the number of available chapters and the contents are not exactly the same with each other. In order to cope with this problem, the necessity of matching the chapters of different writings with each other has emerged. In Table 4.3 information such as average number of words (μ) in matched chapters and their standard deviation (σ) for such averages are provided for all literary works that are used in this study. While giving the statistics, values are rounded to integer whenever it is needed.

Table 4.3: Statistics about used versions of the literary text Leylâ and Mecnun.

	Fuzûlî's Turkish vs. Fuzûlî's English		**Fuzûlî's Turkish vs. Nizamî**	
No. of matched chapters	82		27	
	Fuzûlî's Turkish	**Fuzûlî's English**	**Fuzûlî's Turkish**	**Nizamî**
No. of words	23067	45040	10635	18159
No. of unique words	9981	5680	5409	7717
μ per chapter	283	551	395	674
σ per chapter	205	407	224	280

In order to detect the similar chapters and match them with each other from the different versions of Leylâ and Mecnun, we received help from Bilkent history department graduate students. Together we related the corresponding chapters of these available different versions. By this way, we constructed our ground truth and it is expected to be a handy tool for evaluating the performance of P^2CD. A matched verse from Leylâ and Mecnun works of Fuzûlî in Turkish, in English and Nizamî in Turkish in prose is shown in Figure 4.4.

In addition to statistics of chapters that are given in Table 4.3, to present some reflections about the language we also provide the most frequent words of these literary works in Table A.2. Although all the Leylâ and Mecnun writings are based on the first original literary work, due to they are written by different authors and their written time period differs from each other, there is no perfect

Mecnûn ki haberden oldı âgâh Sûz-ı ciger ile çekdi bir âh Kim gulgulesin hem ol zamanda Cânân eşitdi ol cihânda	Now Mejnun, hearing all this heavy news Drew such a sigh from out his burning heart That all its clamour in the higher world, Where Leyla held her seat, was clearly heard.
Biçare Mecnun Leylâ'nın ölümünü duyunca acı acı ağladı. Bu dünyada kim vardı ki acı acı ağlamamış olsun.	

Figure 4.4: A matched verse from Leylâ and Mecnun works of Fuzûlî in Turkish (top left segment), in English (top right segment) and Nizamî in Turkish in prose (lower segment).

match between the texts. For this reason, while evaluating P^2CD, common chapters which are successfuly matched between the different versions are taken into consideration.

Chapter 5

Experimental Evaluation

In this chapter, firstly the evaluation measures that are used to measure the performance of the proposed external plagiarism and parallel corpus detection algorithm will be discussed in Section 5.1. Then in section 5.2, 5.3 and 5.4 the experimental results of P^2CD will be reported.

5.1 Evaluation Measures

As it is stated before in the previous chapter, P^2CD is mainly tested over PAN'09 which contains 3000 source and 300 suspicious documents. In the evaluation phase, P^2CD's performance is measured by the measures that are already specified by the PAN'09 competition organisers. The success of P^2CD in this study as well as the other methods that compete in PAN'09 competition can be represented as an overall score by making use of the terms like precision, recall, F-measure and granularity. Formulation details of these concepts are provided below. Note that precision, recall and F-measure are well-known measures for evaluating performance. However, granularity is a new concept that is introduced by the competition organisers and it represents if the detected plagiarism are found as a whole (which is the best case and in this case granularity will be equal to 1) or in separate parts (granularity > 1) [2].

$$\text{Recall} = \frac{1}{|S|}\sum_{i=1}^{|S|}\left(\frac{\#\text{ of detected chars of } \mathrm{s_i}}{|\mathrm{s_i}|}\right) \tag{5.1}$$

$$\text{Precision} = \frac{1}{|R|}\sum_{i=1}^{|R|}\left(\frac{\#\text{ of plagiarized chars of } \mathrm{r_i}}{|\mathrm{r_i}|}\right) \tag{5.2}$$

$$\text{Granularity} = \frac{1}{|S_R|}\sum_{i=1}^{|S_R|}(\#\text{ of detections of } \mathrm{s_i} \text{ in R}) \tag{5.3}$$

$$\text{Overall} = \frac{\text{F(harmonic mean of precision and recall)}}{\log_2(1+\text{granularity})} \tag{5.4}$$

In the above formulas, s denotes plagiarized section within the set of all plagiarized sections S whereas r is the found plagiarized section by P^{2}CD within the set of all found plagiarized sections R. S_R stands for found plagiarized sections by P^{2}CD that exist both in S and R. $|s|$ and $|r|$ represents the character lengths of s and r where as $|S|$, $|R|$ and $|S_R|$ represents the size lengths of the respective sets. An example usage of evaluation measures is displayed in Figure 5.1 by assuming block size as seven words.

Plagiarized Passage: The term information retrieval (IR) is equivalent to document retrieval.

Suspicious Text: In this paper, the term information retrieval (IR) is equivalent to document retrieval.

Block 1: In this paper, the term information retrieval

Block 2: (IR) is equivalent to document retrieval.

where $|S| = 1, |R| = 2, |S_R| = 1$

Recall $= \frac{1}{1} \times (\frac{30}{71} + \frac{41}{71}) = 1$, ***Precision*** $= \frac{1}{2} \times (\frac{30}{45} + \frac{41}{41} = 0.84)$, $\boldsymbol{F} = \frac{2\times1.0\times0.84}{1.0+0.84} = 0.92$,

Granularity $= \frac{1}{1} \times (1+1) = 2$, ***Overall*** $= \frac{0.92}{log_2(1+2)} = 0.59$

Figure 5.1: An example usage of evaluation measures.

5.2 Evaluation Results

In this section, the proposed external plagiarism and parallel corpora detection algorithm is evaluated over PAN'09 plagiarism dataset, literary works of Leylâ and Mecnun by different authors or its translation and Bilkent Information Retrieval Group near-duplicate news dataset.

5.2.1 PAN'09 Dataset

We used PAN'09 dataset to measure the plagiarism detection performance of P^2CD. In the next sections we report P^2CD's detection performance and its comparison with Levenstein distance metric which is a well-known existing method for detecting plagiarism cases.

5.2.1.1 Detection of Optimal Parameters

In the experiments over PAN'09 dataset we first focused on optimizing P^2CD parameters. To be more specific, we first try to optimize

1. Block size
2. Step size
3. Cluster count similarity (the difference between the cluster counts of source and suspicious documents in terms of percentage)
4. The similarity between the actual distribution average and Yao distribution average (the difference between the averages in terms of percentage)
5. Gap threshold
6. Consecutiveness threshold.

parameters before running P^{2}CD over the actual dataset (Abbreviation of these parameters as well as some other abbreviations that are used throughout this study can be found in Table A.1). In order to accomplish this, we randomly selected 10 suspicious documents from dataset (they are not included within our PAN'09 sample dataset which contains 300 suspicious documents) and ran P^{2}CD to find the optimal parameter values. Note that 10 documents may sound too low but the size of suspicious and source documents that are used in the experiments are book-sized very large documents. Moreover our proposed algorithm has plenty of components and contains many parameters to be tuned and these are the other reasons that forced us to make the initial tests over a small dataset.

The first parameters that we tried to optimize were block size (bs) and step size (ss) values. Note that the experiments are not conducted for detecting optimal document size. Considering the different document size values (5 words vs. 10 words), 10 words is observed to give the better empirical results. Note that document size cannot be as large as a block size. Since blocks are made up of documents, in case of large documents, blocks will become much larger and this will negatively affect P^{2}CD about effectiveness results. The tests are done for bs: 100, 200, 300 and 500 words and for ss: 1, 3, 5, 10 words. Results of the test can be found in Table 5.1. It is observed that the optimal values for these parameters were when bs: 300 words and ss: 3 words.

Another experiment is conducted that aim to understand if a selected suspicious and source blocks pair are similar to each other about cluster counts. Difference between the cluster counts (n_c) in percentage is calculated as

$$\text{n}_\text{c}\text{ Diff. Percentage} = \frac{Max(n_cSusp, n_cSrc) - Min(n_cSusp, n_cSrc)}{Max(n_cSusp, n_cSrc)} \times 100 \quad (5.5)$$

In the above formula, n_cSusp stands for the cluster count of the suspicious block and n_cSrc stands for the cluster count of the source block. The test is done for cluster similarity threshold values (β) 1%, 5%, 10% and 20%. The results are displayed in Table 5.2. The best performing threshold is found to be 20%.

Table 5.1: Evaluation results of the P^{2}CD on PAN'09 dataset for different block size and step size values.

Block Size	**Measure**	**Step Size=1**	**Step Size=3**	**Step Size=5**	**Step Size=10**
100	Precision	0.0052	0.0058	0.0044	0.0059
	Recall	0.5899	0.5140	0.2071	0.2599
	F-measure	0.0104	0.0115	0.0085	0.01149
	Granularity	5.1000	7.0000	3.8000	1.9000
	Overall	0.0040	0.0040	0.0038	0.0075
200	Precision	0.0063	0.0221	0.0179	0.0157
	Recall	0.5666	0.3809	0.2272	0.3278
	F-measure	0.0123	0.0417	0.0331	0.0300
	Granularity	4.8572	1.5000	2.3000	1.4000
	Overall	0.0049	0.0316	0.0192	0.0238
300	Precision	0.0009	0.0202	0.0182	0.0174
	Recall	0.5909	0.2982	0.1694	0.1485
	F-measure	0.0018	0.0378	0.0329	0.0311
	Granularity	4.0000	1.1000	1.9000	1.5000
	Overall	0.0008	**0.0353**	0.0214	0.0236
500	Precision	0.0034	0.0138	0.0187	0.0250
	Recall	0.0910	0.2000	0.0847	0.0512
	F-measure	0.0065	0.0257	0.0306	0.0337
	Granularity	1.0000	1.0000	1.2000	1.5000
	Overall	0.0065	0.0257	0.0269	0.0255

After finding the optimal values for block size, step size and cluster similarity threshold parameters, another experiment is conducted to find the optimal difference threshold between the actual distribution average and Yao distribution average. The difference percentage between the actual distributon average (AVG_{act}) and Yao distribution average (AVG_{yao}) is calculated as

$$\text{AVG Diff. Percentage} = \frac{AVG_{yao} - AVG_{act}}{AVG_{yao}} \times 100 \tag{5.6}$$

The test is done for similarity threshold values (θ) 1%, 5%, 10% and 20%. The results are displayed in Table 5.3. According to the results, precision increases with the threshold value in proportion and the best performing threshold is found to be 20%.

Table 5.2: Evaluation results of P^{2}CD on PAN'09 dataset for different cluster similarity threshold (β) values.

Block Size	**Step Size**	**Measure**	β**=1%**	β**=5%**	β**=10%**	β**=20%**
		Precision	0.0249	0.0241	0.0240	0.0238
		Recall	0.1862	0.2499	0.2982	0.3997
300	3	F-measure	0.0440	0.0440	0.0446	0.0450
		Granularity	1.3000	1.1000	1.1000	1.1000
		Overall	0.0367	0.0412	0.0417	**0.0421**

Table 5.3: Evaluation results of P^{2}CD on PAN'09 dataset for different difference threshold (θ) values between the actual distribution average and Yao distribution average.

Block Size	**Step Size**	**Measure**	θ**=1%**	θ**=5%**	θ**=10%**	θ**=20%**
		Precision	0.0005	0.0120	0.0770	0.2403
		Recall	0.8021	0.3997	0.2982	0.2850
300	3	F-measure	0.0009	0.0368	0.1224	0.2608
		Granularity	2.0000	1.1000	1.1000	1.5000
		Overall	0.0006	0.0344	0.1144	**0.1973**

The fifth parameter that needs to be optimized was gap threshold (γ) that exists in postprocessing step. The experiment is done for amount of blocks 1, 2, 3, 4 and 5. The results are reported in Table 5.4. According to the results, the best performance for this parameter is observed when gap threshold equals to 2 blocks. Gap threshold is a parameter that is expected to reduce the granularity since it neglects the gaps between the blocks by considering the threshold value. And according to the results, the granularity decreases when we increase gap threshold hence the results meet the expectations.

Table 5.4: Evaluation results of P^{2}CD on PAN'09 dataset for different gap threshold (γ) values.

Block Size	**Step Size**	**Measure**	γ**=1**	γ**=2**	γ**=3**	γ**=4**	γ**=5**
		Precision	0.2403	0.2627	0.2026	0.2026	0.2026
		Recall	0.2850	0.3000	0.3000	0.3000	0.3000
300	3	F-measure	0.2608	0.2802	0.2419	0.2419	0.2419
		Granularity	1.5000	1.1000	1.0000	1.0000	1.0000
		Overall	0.1973	**0.2617**	0.2419	0.2419	0.2419

Another parameter of the postprocessing step is consecutiveness threshold (δ) which is also the last parameter that needs to be optimized. The experiment is done for amount of blocks 2, 3, 4, 5 and 6. The results are reported in Table 5.5. This parameter is expected to increase the precision when we increase the the threshold because long sequence of consecutive blocks are unlikely to occur by chance. And according to the results, precision increases in proportion with the threshold values which meets the expectations. The best performance for this parameter is observed when consecutiveness threshold equals to 3 blocks.

Table 5.5: Evaluation results of P^{2}CD on PAN'09 dataset for different consecutiveness threshold (δ) values.

Block Size	**Step Size**	**Measure**	δ**=2**	δ**=3**	δ**=4**	δ**=5**	δ**=6**
		Precision	0.2627	0.3286	0.3278	0.4916	0.4916
		Recall	0.3000	0.3000	0.2000	0.2000	0.2000
300	3	F-measure	0.2802	0.3137	0.2485	0.2844	0.2844
		Granularity	1.1000	1.1000	1.1000	1.1000	1.1000
		Overall	0.2617	**0.2930**	0.2321	0.2657	0.2657

After conducting the experiments that are given above in details, it is observed that P^{2}CD is worked at best when block size=300 words, step size=3 words, cluster similarity threshold=20%, difference threshold values between the actual distribution average and Yao distribution average=20%, gap threshold=2 blocks and consecutiveness threshold=3 blocks.

As it is discussed in the previous chapter, the actual experiment on PAN'09 dataset with optimal configuration of P^{2}CD is made over 300 documents which includes 100 documents from each obfuscation levels. The average results that are obtained by P^{2}CD for both type of plagiarism cases can be seen in Table 5.6.

As it can also be seen from the above table, P^{2}CD works quite good when the plagiarism case is not obfuscated (all of the derived results for the plagiarized documents with no obfuscation, low obfuscation and high obfuscation can be found in Tables A.3, A.4 and A.5). However, when the plagiarism case is low or highly obfuscated, P^{2}CD fails at detection. In our case, this result is not very surprising because P^{2}CD is only able to detect the plagiarism cases that are structurally

Table 5.6: Evaluation results of P^{2}CD on PAN'09 dataset for different obfuscation levels.

Measure	**No obfuscation**	**Low obfuscation**	**High obfuscation**
Precision	0.7525	0.0402	0.0014
Recall	0.6574	0.0161	0.0018
F-measure	0.7017	0.0230	0.0016
Granularity	1.0800	1.6600	1.1200
Overall	0.6641	0.0163	0.0015

similar but the plagiarism cases which are low or highly obfuscated do not preserve the writing structure of original text. The obtained average overall score for all 300 suspicious documents by P^{2}CD is (0.6641+0.0163+0.0015)/3=0.2273 and with this score we placed ourself to a place between 5th and 6th of the competition (details are given in Table 5.7).

Table 5.7: Performance results of the proposed plagiarism and parallel corpora detection algorithm in comparison with the participants of PAN'09 competition.

Rank	**Overall score**	**F-measure**	**Precision**	**Recall**	**Granularity**	**Participant**
1	0.6957	0.6976	0.7418	0.6585	1.0038	Grozea et al. [16]
2	0.6093	0.6192	0.5573	0.6967	1.0228	Kasprzak et al. [22]
3	0.6041	0.6491	0.6727	0.6272	1.1060	Basile et al. [8]
4	0.3045	0.5286	0.6689	0.4370	2.3317	Yurii Palkovskii [34]
5	***0.2273***	***0.2421***	***0.2647***	***0.2251***	***1.2867***	***P^{2}CD***
6	0.1885	0.4603	0.6051	0.3714	4.4354	Muhr et al. [31]
7	0.1422	0.6190	0.7473	0.5284	19.4327	Scherbinin et al. [39]
8	0.0649	0.1736	0.6552	0.1001	5.3966	Pereira et al.
9	0.0264	0.0265	0.0136	0.4586	1.0068	E. Valls Balaguer [5]
10	0.0187	0.0553	0.0290	0.6048	6.7780	Malcolm et al. [26]
11	0.0117	0.0226	0.3684	0.0116	2.8256	J. Allen

5.2.1.2 Comparison of the P^{2}CD with Levenstein Distance Metric

We used Levenstein distance metric as our baseline method for detecting the plagiarism cases over the same PAN'09 plagiarism dataset portion. Levenstein distance metric is a measure which is also known as edit distance and it is used to measure the similarity between provided two strings. It allows insertions, deletions and substitions within the compared string portions which makes it a handy tool for measuring and detecting the similarity between them [25]. Levenstein distance metric is already adopted in any kind of similarity measuring process

such as DNA analysis [45], speech recognition [33] and plagiarism detection [41]. Moreover, it provides precise results in detecting the similarity cases if the compared strings are in the same language. In PAN'09 competition, the competitor group with the best precision results adapted Levenstein distance metric within their similarity detection method [39].

Before using the Levenstein distance metric in the comparison of suspicious and source document pairs, the compared document pairs are divided into blocks just as in the working process of P^{2}CD. The block size and step size values are kept as the same (bs: 300 words and ss: 3 words) for a healthy comparison of evaluation results. Then the gathered suspicious and source text blocks are compared with each other and in this comparison step, Levenstein distance metric is adopted to measure the distance between the strings that form the suspicious and source blocks. Similar to the case of P^{2}CD, the optimal Levenstein distance value is found over the same randomly selected 10 suspicious documents from PAN'09 dataset. The experiment is done for distance values 100, 60, 30, 15 and 10 character operations. The results are reported in Table 5.8. According to the results, the best performance for the distance parameter is observed when it equals to 15.

Table 5.8: Evaluation results of the the method that uses Levenstein distance for different distance values.

Block Size	**Step Size**	**Measure**	**Distance=100**	**Distance=60**	**Distance=30**	**Distance=15**	**Distance=10**
		Precision	0.0399	0.0767	0.3169	0.3549	0.3654
		Recall	0.3000	0.3000	0.3000	0.3000	0.2768
300	3	F-measure	0.0705	0.1221	0.3083	0.3252	0.3150
		Granularity	1.0000	1.0000	1.0000	1.0000	1.0000
		Overall	0.0705	0.1221	0.3083	**0.3252**	0.3150

As in the case of P^{2}CD, for the method that uses Levenstein distance actual experiment is conducted over the larger part of the PAN'09 dataset. The method is ran over the same 300 randomly selected suspiciuos documents which contains 100 non-obfuscated plagiarism cases, 100 low obfuscated plagiarism cases and 100 high obfuscated plagiarism cases. The average results that are obtained by the method that uses Levenstein distance for both type of plagiarism cases can be seen in Table 5.9.

Table 5.9: Evaluation results of the method that uses Levenstein distance on PAN'09 dataset for different obfuscation levels.

Measure	**No obfuscation**	**Low obfuscation**	**High obfuscation**
Precision	0.7968	0.0430	0.0008
Recall	0.7401	0.0161	0.0010
F-measure	0.7674	0.0234	0.0009
Granularity	1.1200	1.5700	1.1200
Overall	0.7079	0.0172	0.0008

As it can be seen from the above table, similar to the case of P^2CD, the method that uses Levenstein distance gives good results when the plagiarism cases are not confuscated and fails to detect the plagiarism when there is a low or high obfuscation (all of the derived results for the plagiarized documents with no obfuscation, low obfuscation and high obfuscation can be found in Tables A.6, A.7 and A.8). This result meets the expectations because in case of an obfuscation, plagiarized text can be highly modified or paraphrased so the distance between the original text and plagiarized text could result in a very high distance value which cannot be detected as similar by the method that is based on Levenstein distance. The obtained average overall score for all 300 suspicious documents by the method that uses Levenstein distance metric is (0.7079+0.0172+0.0008)/3=0.2420 which is very close to the overall result of P^2CD.

In order to understand if the gathered results by P^2CD and Levenstein distance for different obfuscation levels are similar or show a significant difference, paired t-test experiments are conducted. The experiment is conducted for every type of effectiveness measure (precision, recall, f-measure, granularity, overall score) and the obtained p-values are provided in Table 5.10.

According to the p-values that are provided above, in general for low and high obfuscation the observed performance between the P^2CD and Levenstein distance does not show a significant difference (p-value>0.05). For raw plagiarism case, it is observed that there is a significant difference between the performance results of Levenstein distance and P^2CD. Hence, for raw plagiarism case it can be said that Levenstein performed a better score compared to P^2CD. However, it should be noted that P^2CD is a more comprehensive bilingual method with respect to

Table 5.10: Paired t-test results between P^{2}CD and Levenstein distance for every type of effectiveness measure.

Obfuscation	Method	Precision	Recall	F-Measure	Granularity	Overall
Raw (None)	P^{2}CD	0.7525	0.6574	0.7017	1.0800	0.6641
	Levenstein	0.7968	0.7401	0.7674	1.1200	0.7079
	p-value	0.0131	0.0001	0.0174	0.3484	0.0001
Low	P^{2}CD	0.0402	0.0161	0.0230	1.6600	0.0163
	Levenstein	0.0430	0.0161	0.0234	1.5700	0.0172
	p-value	0.1165	0.4405	0.1542	0.0833	0.0540
High	P^{2}CD	0.0014	0.0018	0.0016	1.1200	0.0015
	Levenstein	0.0008	0.0010	0.0009	1.1200	0.0008
	p-value	0.0584	0.0450	0.2589	1.0000	0.0024

Levenstein distance metric and it is not only designed for plagiarism detection but also parallel corpora detection (results will be provided in the next section). Since Levenstein distance is only designed for measuring string similarities, it is a monolingual method and cannot be used for parallel corpora detection. Moreover, Levenstein distance metric is highly affected from the order of characters within the provided strings and if a complete portion of text is moved to another place within the text, Levenstein distance may treat this action as a series of differences rather than a single text portion move operation [14].

5.2.2 Leylâ and Mecnun Translations

Literary works of Fuzûlî's Leylâ and Mecnun in Turkish, in English and the same literary work by Nizamî in Turkish are used to evaluate the parallel corpora detection of P^{2}CD in this study. In the next 2 sections, P^{2}CD's findings about the comparison of different versions of Leylâ and Mecnun literary works will be reported.

5.2.2.1 Fuzûlî's Turkish Version vs. Fuzûlî's English Version

In order to be able to use these literary works with our proposed external plagiarism and parallel corpora detection algorithm, we accepted each literary work as a single block and their chapters as the documents of this block. However, P^{2}CD requires the document counts of the compared blocks to be equal. Document counts should be equal because while comparing the cluster distributions of the clustered blocks, P^{2}CD assumes that the document counts of the compared blocks are equal and it tries to find each and every document that exist in suspicious text block clusters within the clusters of source text block. For this reason a chapter matching operation is done between these two corresponding compared literary works. According to this matching operation 82 matching chapters are detected manually between Fuzûlî's Turkish version and its English version. However, during this manual matching process although the titles of the chapters appeared as quite similar in each literary work it is observed that the chapters are only slightly similar to each other. P^{2}CD is then ran for these two blocks with 82 documents and the actual distribution vs. Yao distribution results that are found by P^{2}CD are given in Table 5.11. In the process of parallel corpora detection, some parameters of P^{2}CD which aim to reduce the number of blocks that needs to be compared (cluster count similarity, the similarity between the actual distribution and Yao distribution) and the ones which are used in postprocessing step (gap threshold, consecutiveness threshold) and needs more than one blocks present are ignored since each of the compared literary works is accepted as a single block.

Table 5.11: Actual distribution vs. Yao distribution results that are found by P^{2}CD for the literary works Fuzûlî's Turkish and Fuzûlî's English by considering chapters as documents.

Actual Distribution	Yao Distribution
1.4048	1.4467

The result shows us that the cluster distributions of the literary works look more similar to each other than the random case. However, the difference from the random (1.41 vs. 1.45) is not very high which can be seen as a sign that these compared literary works do not look like significantly similar to each other from random. Hence, the found result of P^{2}CD suits with our observations. In order to

verify the result Monte Carlo experiments are performed (results are provided in Figure 5.2). Monte Carlo experiments define a reference population and provide a baseline distribution [19]. In all Monte Carlo experiments we generate 1000 random cases of cluster distributions. According to Monte Carlo experiment, the actual distribution score is lower than the 54.3% of the randomly obtained distribution cases.

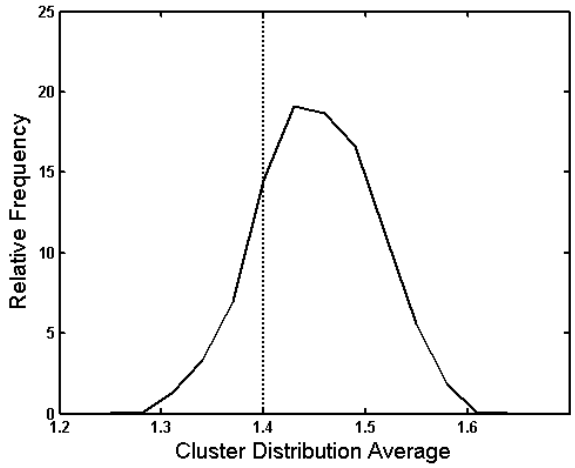

Figure 5.2: Distribution of Monte Carlo values for actual distribution of Fuzûlî's Turkish (by considering chapters as documents) that is found by P^{2}CD.

As it is mentioned before, in the matching process we realized that the titles of the chapters look like more similar rather than the content of chapters. For this reason we conducted another experiment in which again we accept each literary work as single block but this time titles of the 82 chapters as documents rather than the content of chapters. The actual distribution vs. Yao distribution results that are found by P^{2}CD for the chapters of the compared literary works are given in Table 5.12.

Table 5.12: Actual distribution vs. Yao distribution results that are found by P^{2}CD for the literary works Fuzûlî's Turkish and Fuzûlî's English by considering chapter titles as documents.

Actual Distribution	**Yao Distribution**
1.7778	1.9193

The results for chapter titles are just as expected and there is a significant difference between the actual distribution and the random distribution cases. In order to verify the results again Monte Carlo experiments are performed. The results can be seen in Figure 5.3. According to Monte Carlo experiment, the actual distribution score is lower than the 97.4% of the randomly obtained distribution cases.

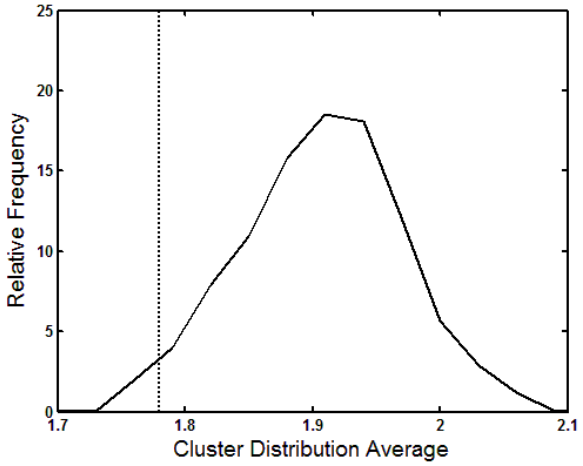

Figure 5.3: Distribution of Monte Carlo values for actual distribution of Fuzûlî's Turkish (by considering titles as documents) that is found by the P^{2}CD.

5.2.2.2 Fuzûlî's Turkish Version vs. Nizamî

The same experiment is also conducted between Fuzûlî's Leylâ and Mecnun Turkish version and the one that is written by Nizamî in Turkish. Again a chapter matching operation is done between these two corresponding compared literary works. According to this matching operation 27 matching chapters are detected manually between Fuzûlî's Leylâ and Mecnun Turkish version and the one that is written by Nizamî. During this manual matching process it is observed that these two compared literary works do not show any similarity to each other. Then P^{2}CD is then ran for these two blocks with 27 documents and the actual distribution vs. Yao distribution results that are found by P^{2}CD are given in Table 5.13.

Table 5.13: Actual distribution vs. Yao distribution results that are found by P^{2}CD for the literary works Fuzûlî's Leylâ and Mecnun in Turkish and Nizamî's Leylâ and Mecnun by considering chapters as documents.

Actual Distribution	Yao Distribution
1.5295	1.5209

According to the cluster distribution results, there is no difference between the random case and the actual distribution case which shows that the compared texts do not show any meaningful similarity. Hence, once again our insight about the compared literary works are verified by P^{2}CD. Monte Carlo experiment results for the compared literary works can be seen in Figure 5.4. According to Monte Carlo experiment, the actual distribution score is only lower than the 23.3% of the randomly obtained distribution cases.

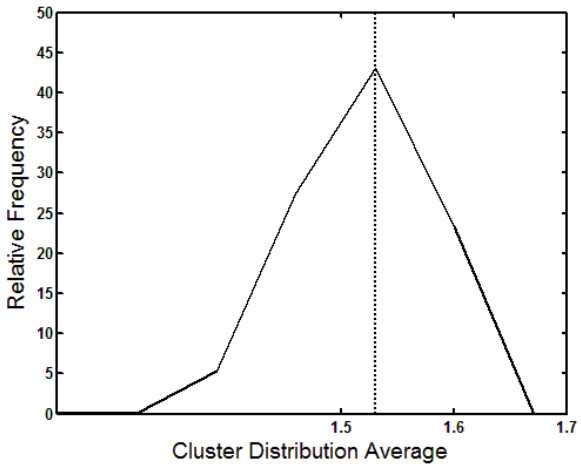

Figure 5.4: Distribution of Monte Carlo values for actual distribution of Fuzûlî's Leylâ and Mecnun in Turkish over Nizamî's Leylâ and Mecnun in Turkish that is found by P^{2}CD.

Chapter 6

Conclusion

In this study we presented a novel external plagiarism and parallel corpora detection method. Our method P^2CD is making use of the structural similarities of the compared texts in the detection process and is language independent. The plagiarism detection evaluation of P^2CD is tested over PAN'09 plagiarism dataset. Initially, parameters of P^2CD are optimized over a small test dataset that is composed of documents which are randomly selected from PAN'09 dataset. Later the actual plagiarism detection evaluation is done over a larger set of 300 randomly selected documents which contains 100 raw plagiarism (no obfuscation), 100 low obfuscation and 100 high obfuscation cases. We observed that P^2CD gives promising results in detecting raw plagiarism cases but failed to detect the most of the existing plagiarism cases when the obfuscation is present. From our side, this result was not surprising because our method is only making use of structural similarities between the texts while deciding if they are similar without using any other knowledge such as semantic or syntactic information. In case of an obfuscation, this structural similarity between the texts are not preserved and P^2CD gives poor results. P^2CD is then compared with Levenstein distance as baseline method which is a well-known existing technique that is used in plagiarism detection. According to paired t-test results, both of the compared methods showed a similar poor performance in detecting plagiarism cases with low or high obfuscation. For the case of raw plagiarism, we observed Levenstein distance

showed slightly better overall performance. However, P^{2}CD is not only designed for plagiarism detection but also for parallel corpora detection. It can be used in detecting bilingual textual similarities unlike Levenstein distance which only measures the number of operations (insertion, deletion and substition) needed to transform one string into another hence can only give good results for monolingual corpus. The parallel corpora detection evaluation of P^{2}CD is done over Fuzûlî's Leylâ and Mecnun which is rewritten by many authors by the time. Initially chapters of the compared literary works are matched manually. Then each compared literary work is accepted as a single block and its chapters are accepted as documents in order to run our method succesfully. According to the results, P^{2}CD is able to detect if the compared literary work pairs possess a structural similarity. Likewise, if they don't possess any structural similarity in reality, observed results of P^{2}CD clearly shows that they do not carry any similarity.

P^{2}CD is an unsupervised clustering technique and it is solely based on detecting structural similarities. Hence, currently it is not able to detect documents having semantic similarity. P^{2}CD is further evaluated over Bilkent Information Retrieval Group near-duplicate dataset [44] which contains semantically similar documents and the obtained poor performance also confirmed this assumption.

As future work, components of P^{2}CD can be further investigated. For example, in clustering step a different clustering algorithm such as k-means can be used. Similarly in indexing step a different indexing scheme can be adopted. Also, in selecting candidate documents step, instead of using n-gram distance, an existing indexing tool or fingerprinting technique can be useful for increasing the overall performance of P^{2}CD.

Bibliography

[1] Online etymology dictionary. http://www.etymonline.com/index.php?term=plagiarism, 2011.

[2] Pan 2009. http://www.uni-weimar.de/medien/webis/research/workshopseries/pan-09/competition.html, 2011.

[3] A. Abbasi and H. Chen. Visualizing authorship for identification. In *ISI*, pages 60–71, 2006.

[4] A. Abbasi and H. Chen. Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace. *ACM Trans. Inf. Syst.*, 26(2):7:1–7:29, April 2008.

[5] E. V. Balaguer. Putting ourselves in sme's shoes: Automatic detection of plagiarism by the wcopyfind tool. PAN'09 Workshop, 2009.

[6] C. Bannard and C. Callison-Burch. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 597–604, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[7] R. Barzilay and K. R. McKeown. Extracting paraphrases from a parallel corpus. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, ACL '01, pages 50–57, Stroudsburg, PA, USA, 2001. Association for Computational Linguistics.

[8] C. Basile, D. Benedetto, E. Caglioti, G. Cristadoro, and M. Degli Esposti. A plagiarism detection procedure in three steps: Selection, matches and "squares". PAN'09 Workshop, 2009.

[9] C. Basile, D. Benedetto, E. Caglioti, and M. D. Esposti. An example of mathematical authorship attribution. *Journal of Mathematical Physics*, 49(12):13–19, 2008.

[10] Y. Bernstein and J. Zobel. A scalable system for identifying co-derivative documents. In *In Proceedings of the Symposium on String Processing and Information Retrieval*, pages 55–67. Springer, 2004.

[11] F. Can, E. F. Can, and C. Karbeyaz. Translation relationship quantification: A cluster-based approach and its application to Shakespeare's sonnets. In *In the Proceedings of the 25th International Symposium on Computer and Information Sciences (ISCIS'10)*, pages 117–120. Springer, 2010.

[12] F. Can, S. Kocberber, O. Baglioglu, S. Kardas, H. C. Ocalan, and E. Uyar. New event detection and topic tracking in turkish. *Journal of the American Society for Information Science and Technology*, 61(4):802–819, 2010.

[13] F. Can and E. A. Ozkarahan. Concepts and effectiveness of the cover-coefficient-based clustering methodology for text databases. *ACM Trans. Database Syst.*, 15(4):483–517, December 1990.

[14] P. Clough. Plagiarism in natural and programming languages: an overview of current tools and technologies. Technical report, University of Sheffield, 2000.

[15] M. N. Doğan. *Leylâ ve Mecnun*. Yapı Kredi Yayınları, İstanbul, 2000.

[16] C. Grozea, C. Gehl, and M. Popescu. ENCOPLOT: Pairwise Sequence Matching in Linear Time Applied to Plagiarism Detection. In *3rd PAN Workshop. Uncovering Plagiarism, Authorship and Social Software Misuse*, page 10, 2009.

[17] S. Hariharan, S. Kamal, A. V. M. Faisal, S. M. Azharudheen, and B. Raman. Detecting plagiarism in text documents. In *BAIP*, pages 497–500, 2010.

[18] S. Huri. *Leyla and Mejnun.* Allen and Unwin - UNESCO, London, 1970.

[19] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data.* Prentice Hall. Inc., Upper Saddle River, NJ, USA, 1988.

[20] C. Karbeyaz, E. F. Can, F. Can, and M. Kalpaklı. A content-based social network study of Evliyâ Çelebi's *Seyahatnâme-Bitlis Section.* In *In the Proceedings of the 26th International Symposium on Computer and Information Sciences (ISCIS'11).* Springer, 2011.

[21] S. Kardaş. New event detection and tracking in turkish. Master's thesis, Bilkent University, 2009.

[22] J. Kasprzak, M. Brandejs, and M. Kripac. Finding plagiarism by evaluating document similarities. PAN'09 Workshop, 2009.

[23] B. Kjell, W. A. Woods, and O. Frieder. Discrimination of authorship using visualization. *Inf. Process. Manage.*, 30(1):141–150, January 1994.

[24] J. Koberstein and Y.-K. Ng. Using word clusters to detect similar web documents. In J. Lang, F. Lin, and J. Wang, editors, *KSEM*, volume 4092 of *Lecture Notes in Computer Science*, pages 215–228. Springer, 2006.

[25] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966.

[26] J. A. Malcolm and P. C. R. Lane. Tackling the pan'09 external plagiarism detection corpus with a desktop plagiarism detector. PAN'09 Workshop, 2009.

[27] Z. Marx, I. Dagan, J. M. Buhmann, and E. Shamir. Coupled clustering: a method for detecting structural correspondence. *J. Mach. Learn. Res.*, 3:747–780, March 2003.

[28] H. Maurer, F. Kappe, and B. Zaka. Plagiarism - A Survey. *j-jucs*, 12(8):1050–1084, 2006.

[29] B. Mizrahi. Paraphrase extraction from parallel news corpora. Master's thesis, Koc University, 2006.

[30] M. Muhr, R. Kern, M. Zechner, and M. Granitzer. External and intrinsic plagiarism detection using a cross-lingual retrieval and segmentation system - Lab Report for PAN at CLEF 2010. In *CLEF (Notebook Papers/LABs/Workshops)'10*, pages 1–10, 2010.

[31] M. Muhr, M. Zechner, and R. Kern. External and intrinsic plagiarism detection using vector space models. PAN'09 Workshop, 2009.

[32] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443 – 453, 1970.

[33] J. Nerbonne, W. Heeringa, and P. Kleiweg. Edit distance and dialect proximity. In D. Sankoff and J. B. Kruskal, editors, *Time Warps, String Edits, and Macromolecules - The Theory and Practice of Sequence Comparison*, pages 5–15. CSLI Publications, Stanford, CA 94305, 1999.

[34] Y. Palkovskii. Counter plagiarism detection software and Counter counter plagiarism detection methods. PAN'09 Workshop, 2009.

[35] M. Potthast, B. Stein, A. Eiselt, A. Barrn-Cedeo, and P. Rosso. *Overview of the 1st International Competition on Plagiarism Detection*, pages 1–9. CEUR-WS.org, 2009.

[36] P. Resnik and N. A. Smith. The web as a parallel corpus. *Comput. Linguist.*, 29(3):349–380, September 2003.

[37] G. Salton. *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[38] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986.

[39] V. Scherbinin and S. Butakov. Using microsoft sql server platform for plagiarism detection. PAN'09 Workshop, 2009.

[40] S. Schleimer, D. S. Wilkerson, and A. Aiken. Winnowing: local algorithms for document fingerprinting. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, SIGMOD '03, pages 76–85, New York, NY, USA, 2003. ACM.

[41] Z. Su, B.-R. Ahn, K.-Y. Eom, M.-K. Kang, J.-P. Kim, and M.-K. Kim. Plagiarism detection using the Levenshtein distance and Smith-Waterman algorithm. *Innovative Computing ,Information and Control, International Conference on*, 0:569, 2008.

[42] A. N. Tarlan. *Leylâ ve Mecnun*. M.E.B. Yayınları, Ankara, 1989.

[43] C. Vania, M. Adriani, F. I. Komputer, and K. Depok. Automatic external plagiarism detection using passage similarities. PAN'09 Workshop, 2009.

[44] E. Varol, F. Can, C. Aykanat, and O. Kaya. CoDet: Sentence-based containment detection in news corpora. 20th ACM Conference on Information and Knowledge Management (CIKM'11), 2011. (To appear).

[45] J.-H. Xu. Identifying g-protein coupled receptors using weighted levenshtein distance and nearest neighbor method. *Proteomics and Bioinformatics*, 3(4):252–257, 2005.

[46] S. B. Yao. Approximating block accesses in database organizations. *Commun. ACM*, 20(4):260–261, April 1977.

[47] D. Zou, W. jiang Long, and Z. Ling. A cluster-based plagiarism detection method - Lab Report for PAN at CLEF 2010. In M. Braschler, D. Harman, and E. Pianta, editors, *CLEF (Notebook Papers/LABs/Workshops)*, 2010.

Appendix A

Data

Table A.1: Definitions of the symbols used.

Symbol	**Definition**
P^2CD	abbreviation of the proposed external plagiarism and parallel corpora detection method
EPD	external plagiarism detection
PE	paraphrase extraction
bs	block size
ss	step size
ds	document size
n_c	number of clusters
β	cluster similarity threshold
θ	difference percentage between the actual vs. Yao distribution average
γ	gap threshold
δ	consecutiveness threshold

Table A.2: Most frequent words in all versions of the literary text Leylâ and Mecnun.

Fuzûlî's Turkish vs Fuzûlî's English				**Fuzûlî's Turkish vs Nizamî**			
Fuzûlî's Turkish		**Fuzûlî's English**		**Fuzûlî's Turkish**		**Nizamî**	
Word	Frequency	Word	Frequency	Word	Frequency	Word	Frequency
ü	400	the	2384	ü	187	bir	502
ile	303	of	1529	ki	149	bu	370
ki	297	and	1413	bu	144	ve	331
bu	271	to	929	ile	130	gibi	227
bir	206	in	856	ol	103	o	200
oldı	195	a	706	bir	99	senin	131
ol	191	that	645	oldı	95	ki	129
kim	123	my	614	ey	62	her	113
etdi	116	thy	595	etdi	60	ne	110
ne	113	all	519	ola	56	ile	106
ey	106	his	488	ne	53	de	97
idi	103	is	451	kim	52	da	90
ola	101	with	450	idi	47	ben	89
hem	98	i	394	kim	45	için	85
kim	91	for	360	eyle	44	kadar	82
eyle	85	now	348	hem	42	onu	77
ve	72	her	344	sana	42	ey	76
menî	72	thou	339	var	38	onun	74
ana	71	he	281	mânâ	35	sen	72
sana	71	this	266	bir	35	olan	62
bir	69	as	261	kimi	32	fakat	60
mânâ	68	not	257	her	31	benim	59
var	65	from	236	ana	30	böyle	57
her	64	love	233	ol	28	leyla	53
kimi	64	no	226	ve	28	beni	52
vü	58	but	214	eyledi	26	mecnun	51
sen	57	was	204	men	26	çok	49
men	56	be	196	özge	24	daha	45
ol	56	thee	187	et	24	bana	42
u	55	me	181	yoh	23	eğer	42

Table A.3: P^{2}CD's no obfuscation plagiarism results for randomly selected 100 documents used in the experiments.

No	Document No	Precision	Recall	F-Measure	Granularity	Overall
1	3333	0.5878	0.1832	0.2793	1.0000	0.2793
2	3369	0.6843	1.0000	0.8126	2.0000	0.5127
3	6112	1.0000	0.7935	0.8849	1.0000	0.8849
4	3976	0.9816	1.0000	0.9907	1.0000	0.9907
5	5205	0.5868	1.0000	0.7396	1.0000	0.7396
6	5567	0.6257	1.0000	0.7698	2.0000	0.4857
7	5768	1.0000	0.7590	0.8630	1.0000	0.8630
8	11524	1.0000	0.6449	0.7841	2.0000	0.4947
9	2925	0.7826	0.9300	0.8500	1.0000	0.8500
10	3962	0.3425	0.4719	0.3969	1.0000	0.3969
11	10739	1.0000	0.8076	0.8936	1.0000	0.8936
12	2956	0.8793	0.6898	0.7731	1.0000	0.7731
13	5149	0.9849	0.4831	0.6482	1.0000	0.6482
14	4216	0.4865	1.0000	0.6546	1.0000	0.6546
15	4	0.6919	0.0881	0.1563	1.0000	0.1563
16	2233	0.4526	0.8062	0.5797	1.0000	0.5797
17	13160	1.0000	0.4041	0.5756	1.0000	0.5756
18	2772	0.0000	0.0000	0.0000*	1.0000	0.0000
19	4667	0.8157	1.0000	0.8985	1.0000	0.8985
20	10081	0.6140	0.1922	0.2928	1.0000	0.2928
21	14360	1.0000	0.7781	0.8752	1.0000	0.8752
22	11926	0.8765	0.2621	0.4035	1.0000	0.4035
23	3711	0.9902	0.9376	0.9632	1.0000	0.9632
24	12402	0.7963	0.3279	0.4645	1.0000	0.4645
25	10026	0.4976	0.6366	0.5586	1.0000	0.5586
26	2175	0.8000	0.4012	0.5344	1.0000	0.5344
27	12094	0.5711	0.8544	0.6846	1.0000	0.6846
28	190	0.9950	0.6615	0.7947	1.0000	0.7947
29	13450	0.6243	0.7870	0.6963	1.0000	0.6963
30	761	0.9750	0.2767	0.4311	1.0000	0.4311
31	4693	0.7500	1.0000	0.8571	1.0000	0.8571
32	5654	0.3293	0.3291	0.3292	1.0000	0.3292
33	3828	1.0000	0.0904	0.1658	1.0000	0.1658
34	8385	0.6855	0.0893	0.1580	1.0000	0.1580
35	5690	0.9431	0.6679	0.7820	1.0000	0.7820
36	3398	0.3989	0.7204	0.5135	1.0000	0.5135
37	3996	0.9593	0.7861	0.8641	1.0000	0.8641
38	553	0.7547	0.9756	0.8511	1.0000	0.8511
39	6014	0.8019	0.8919	0.8445	1.0000	0.8445
40	3965	1.0000	1.0000	1.0000	1.0000	1.0000
41	12559	0.6228	0.7373	0.6752	1.0000	0.6752
42	8368	0.8484	0.6900	0.7610	1.0000	0.7610
43	122	0.1756	0.9700	0.2974	1.0000	0.2974
44	1468	0.8438	0.6400	0.7279	1.0000	0.7279
45	10420	0.6483	1.0000	0.7866	1.0000	0.7866
46	11905	1.0000	0.0470	0.0897	2.0000	0.0566
47	6384	1.0000	0.4029	0.5744	1.0000	0.5744
48	6560	1.0000	0.2674	0.4220	1.0000	0.4220
49	6267	0.3712	0.2022	0.2618	1.0000	0.2618

* It is counted as 0 since precision and recall values are also 0.

No	Document No	Precision	Recall	F-Measure	Granularity	Overall
50	3477	1.0000	0.8460	0.9166	1.0000	0.9166
51	13157	0.8833	0.9373	0.9095	1.0000	0.9095
52	14050	0.7119	0.6707	0.6907	1.0000	0.6907
53	6341	0.7243	0.6175	0.6666	1.0000	0.6666
54	576	0.5695	1.0000	0.7257	1.0000	0.7257
55	10408	0.9100	0.5048	0.6494	1.0000	0.6494
56	972	0.8956	0.7225	0.7998	1.0000	0.7998
57	14235	0.5639	1.0000	0.7211	1.0000	0.7211
58	13303	0.9861	0.7312	0.8397	1.0000	0.8397
59	2344	0.6212	0.8836	0.7295	1.0000	0.7295
60	14358	1.0000	0.3080	0.4709	1.0000	0.4709
61	8248	0.7478	0.3820	0.5057	2.0000	0.3190
62	3097	0.9903	0.9342	0.9614	1.0000	0.9614
63	12334	0.2900	1.0000	0.4496	1.0000	0.4496
64	1988	1.0000	0.6701	0.8025	1.0000	0.8025
65	2705	0.9526	0.4955	0.6519	1.0000	0.6519
66	917	0.6663	1.0000	0.7997	1.0000	0.7997
67	6575	0.7817	0.1076	0.1892	1.0000	0.1892
68	1729	0.7658	0.5474	0.6384	1.0000	0.6384
69	1166	0.7411	0.8127	0.7753	1.0000	0.7753
70	14401	0.9159	0.5208	0.6640	2.0000	0.4190
71	6194	0.9280	0.4136	0.5722	1.0000	0.5722
72	6026	0.6087	0.8750	0.7179	1.0000	0.7179
73	2025	0.9589	0.9795	0.9691	1.0000	0.9691
74	8564	0.5227	1.0000	0.6866	1.0000	0.6866
75	12044	0.7155	0.9375	0.8116	1.0000	0.8116
76	2327	0.8214	1.0000	0.9019	1.0000	0.9019
77	4982	0.6896	0.3030	0.4210	1.0000	0.4210
78	3011	0.1530	1.0000	0.2654	1.0000	0.2654
79	1469	0.7065	0.6669	0.6861	1.0000	0.6861
80	1373	0.8086	1.0000	0.8942	1.0000	0.8942
81	4658	0.8900	0.8906	0.8903	2.0000	0.5617
82	5701	0.7119	0.5955	0.6485	1.0000	0.6485
83	2725	0.4590	0.5284	0.4913	1.0000	0.4913
84	3406	0.6901	0.5206	0.5935	1.0000	0.5935
85	11927	0.9960	0.5490	0.7078	1.0000	0.7078
86	14231	0.1827	0.8719	0.3021	1.0000	0.3021
87	1985	0.5611	0.8129	0.6639	1.0000	0.6639
88	1135	0.8721	0.0028	0.0056	2.0000	0.0035
89	1606	0.7110	0.7511	0.7305	1.0000	0.7305
90	8622	0.9465	0.7898	0.8611	1.0000	0.8611
91	11720	0.9148	0.9048	0.9098	1.0000	0.9098
92	135	0.9685	0.3925	0.5586	1.0000	0.5586
93	6695	0.6045	1.0000	0.7535	1.0000	0.7535
94	6209	0.3747	0.5984	0.4608	1.0000	0.4608
95	3785	0.8331	0.6316	0.7185	1.0000	0.7185
96	4257	0.5739	0.6253	0.5985	1.0000	0.5985
97	6512	0.8473	0.2142	0.3420	1.0000	0.3420
98	12024	1.0000	0.0100	0.0198	1.0000	0.0198
99	12530	1.0000	0.6668	0.8001	1.0000	0.8001
100	6227	0.7048	0.8309	0.7627	1.0000	0.7627
	Average	0.7525	0.6574	0.7017	1.0800	0.6641

Table A.4: P^{2}CD's low obfuscation plagiarism results for randomly selected 100 documents used in the experiments.

No	Document No	Precision	Recall	F-Measure	Granularity	Overall
1	4610	0.0811	0.0097	0.0173	2.0000	0.0109
2	825	0.0059	0.0013	0.0021	2.0000	0.0013
3	6103	0.0332	0.0242	0.0280	2.0000	0.0177
4	3740	0.0000	0.0000	0.0000*	1.0000	0.0000
5	2967	0.0355	0.0191	0.0248	2.0000	0.0157
6	6626	0.0854	0.0092	0.0166	2.0000	0.0105
7	6158	0.0610	0.0350	0.0445	1.0000	0.0445
8	8602	0.0580	0.0076	0.0134	2.0000	0.0085
9	3967	0.0378	0.0172	0.0236	1.0000	0.0236
10	5007	0.0000	0.0000	0.0000*	1.0000	0.0000
11	1851	0.0959	0.0207	0.0341	2.0000	0.0215
12	2637	0.0815	0.0210	0.0334	2.0000	0.0211
13	5513	0.0493	0.0220	0.0304	2.0000	0.0192
14	2865	0.0266	0.0433	0.0330	2.0000	0.0208
15	13425	0.0361	0.0150	0.0212	2.0000	0.0134
16	8384	0.0000	0.0000	0.0000*	1.0000	0.0000
17	8871	0.1355	0.0241	0.0409	2.0000	0.0258
18	1062	0.0480	0.0213	0.0295	2.0000	0.0186
19	3512	0.1045	0.0002	0.0004	2.0000	0.0003
20	1829	0.0000	0.0000	0.0000*	1.0000	0.0000
21	4569	0.0328	0.0183	0.0235	2.0000	0.0148
22	382	0.0161	0.0264	0.0200	1.0000	0.0200
23	12048	0.0561	0.0067	0.0120	2.0000	0.0076
24	12118	0.0349	0.0366	0.0357	2.0000	0.0225
25	5872	0.0510	0.0144	0.0225	2.0000	0.0142
26	1086	0.0151	0.0197	0.0171	2.0000	0.0108
27	8747	0.0000	0.0000	0.0000*	1.0000	0.0000
28	3752	0.0555	0.0153	0.0240	2.0000	0.0151
29	2857	0.0357	0.0189	0.0247	2.0000	0.0156
30	1751	0.0568	0.0097	0.0166	1.0000	0.0166
31	8046	0.0529	0.0130	0.0209	2.0000	0.0132
32	2416	0.0563	0.0212	0.0308	2.0000	0.0194
33	3980	0.0421	0.0126	0.0194	2.0000	0.0122
34	3820	0.0010	0.0185	0.0019	2.0000	0.0012
35	8323	0.0000	0.0000	0.0000*	1.0000	0.0000
36	11949	0.0000	0.0000	0.0000*	1.0000	0.0000
37	3898	0.0682	0.0326	0.0441	1.0000	0.0441
38	13199	0.0119	0.0308	0.0172	2.0000	0.0108
39	2791	0.0082	0.0129	0.0100	2.0000	0.0063
40	14135	0.0432	0.0170	0.0244	2.0000	0.0154
41	4162	0.0255	0.0314	0.0281	2.0000	0.0178
42	3771	0.0000	0.0000	0.0000*	1.0000	0.0000
43	2447	0.0187	0.0206	0.0196	1.0000	0.0196
44	14342	0.0647	0.0203	0.0309	2.0000	0.0195
45	10657	0.0000	0.0000	0.0000*	1.0000	0.0000
46	3859	0.0004	0.0266	0.0008	2.0000	0.0005
47	260	0.0897	0.0120	0.0212	2.0000	0.0134
48	8187	0.0000	0.0000	0.0000*	1.0000	0.0000
49	14133	0.0795	0.0029	0.0056	2.0000	0.0035

* It is counted as 0 since precision and recall values are also 0.

No	Document No	Precision	Recall	F-Measure	Granularity	Overall
50	2451	0.0569	0.0236	0.0334	2.0000	0.0210
51	2068	0.0733	0.0234	0.0355	2.0000	0.0224
52	968	0.0821	0.0145	0.0246	2.0000	0.0156
53	1850	0.0000	0.0000	0.0000*	1.0000	0.0000
54	4285	0.0327	0.0147	0.0203	2.0000	0.0128
55	3751	0.0000	0.0000	0.0000*	1.0000	0.0000
56	1541	0.0321	0.0316	0.0318	1.0000	0.0318
57	5764	0.0501	0.0192	0.0278	2.0000	0.0175
58	4430	0.0255	0.0093	0.0136	2.0000	0.0086
59	14367	0.0292	0.0221	0.0252	2.0000	0.0159
60	2173	0.0763	0.0287	0.0417	2.0000	0.0263
61	10031	0.0499	0.0442	0.0469	2.0000	0.0296
62	434	0.0615	0.0297	0.0401	1.0000	0.0401
63	4904	0.0550	0.0039	0.0073	2.0000	0.0046
64	111	0.1163	0.0154	0.0272	2.0000	0.0172
65	11921	0.0703	0.0187	0.0295	1.0000	0.0295
66	3044	0.0389	0.0103	0.0163	2.0000	0.0103
67	4668	0.0356	0.0023	0.0043	2.0000	0.0027
68	12056	0.0000	0.0000	0.0000*	1.0000	0.0000
69	460	0.0603	0.0195	0.0295	2.0000	0.0186
70	8504	0.0868	0.0345	0.0494	1.0000	0.0494
71	13386	0.0725	0.0272	0.0396	2.0000	0.0250
72	925	0.0000	0.0000	0.0000*	1.0000	0.0000
73	3297	0.0291	0.0414	0.0342	2.0000	0.0216
74	5109	0.1105	0.0275	0.0440	2.0000	0.0278
75	13235	0.0000	0.0000	0.0000*	1.0000	0.0000
76	6177	0.0724	0.0123	0.0210	2.0000	0.0133
77	8968	0.0660	0.0109	0.0187	2.0000	0.0118
78	14142	0.0000	0.0000	0.0000*	1.0000	0.0000
79	10553	0.0408	0.0337	0.0369	2.0000	0.0233
80	11867	0.0421	0.0087	0.0144	2.0000	0.0091
81	13093	0.0538	0.0198	0.0289	2.0000	0.0183
82	6171	0.0334	0.0295	0.0313	2.0000	0.0198
83	3592	0.0738	0.0322	0.0448	2.0000	0.0283
84	8991	0.0388	0.0225	0.0285	2.0000	0.0180
85	3776	0.0297	0.0229	0.0259	2.0000	0.0163
86	1267	0.0667	0.0131	0.0219	2.0000	0.0138
87	8406	0.0463	0.0297	0.0362	2.0000	0.0228
88	6006	0.0000	0.0000	0.0000*	1.0000	0.0000
89	198	0.0000	0.0000	0.0000*	1.0000	0.0000
90	5038	0.0397	0.0363	0.0379	2.0000	0.0239
91	6687	0.0000	0.0000	0.0000*	1.0000	0.0000
92	2969	0.0000	0.0000	0.0000*	1.0000	0.0000
93	13200	0.0192	0.0299	0.0234	2.0000	0.0148
94	8039	0.0688	0.0457	0.0549	2.0000	0.0347
95	3025	0.0500	0.0043	0.0079	1.0000	0.0079
96	8356	0.0577	0.0150	0.0238	2.0000	0.0150
97	6362	0.0299	0.0408	0.0345	1.0000	0.0345
98	3912	0.0000	0.0000	0.0000*	1.0000	0.0000
99	1443	0.0000	0.0000	0.0000*	1.0000	0.0000
100	294	0.0517	0.0125	0.0201	2.0000	0.0127
	Average	0.0402	0.0161	0.0230	1.6600	0.0163

* It is counted as 0 since precision and recall values are also 0.

Table A.5: P^{2}CD's high obfuscation plagiarism results for randomly selected 100 documents used in the experiments.

No	Document No	Precision	Recall	F-Measure	Granularity	Overall
1	10608	0.0000	0.0000	0.0000*	1.0000	0.0000
2	10398	0.0000	0.0000	0.0000*	1.0000	0.0000
3	2345	0.0086	0.0079	0.0082	2.0000	0.0052
4	8870	0.0101	0.0102	0.0101	2.0000	0.0064
5	5642	0.0162	0.0125	0.0141	1.0000	0.0141
6	10617	0.0000	0.0000	0.0000*	1.0000	0.0000
7	4095	0.0000	0.0000	0.0000*	1.0000	0.0000
8	6207	0.0097	0.0114	0.0105	2.0000	0.0066
9	12023	0.0000	0.0000	0.0000*	1.0000	0.0000
10	6569	0.0086	0.0143	0.0107	2.0000	0.0068
11	2289	0.0000	0.0000	0.0000*	1.0000	0.0000
12	1853	0.0000	0.0000	0.0000*	1.0000	0.0000
13	14290	0.0000	0.0000	0.0000*	1.0000	0.0000
14	919	0.0000	0.0000	0.0000*	1.0000	0.0000
15	3639	0.0000	0.0000	0.0000*	1.0000	0.0000
16	2443	0.0000	0.0000	0.0000*	1.0000	0.0000
17	3853	0.0064	0.0093	0.0076	2.0000	0.0048
18	458	0.0000	0.0000	0.0000*	1.0000	0.0000
19	6497	0.0000	0.0000	0.0000*	1.0000	0.0000
20	5089	0.0000	0.0000	0.0000*	1.0000	0.0000
21	13359	0.0000	0.0000	0.0000*	1.0000	0.0000
22	2054	0.0000	0.0000	0.0000*	1.0000	0.0000
23	12263	0.0111	0.0092	0.0101	2.0000	0.0063
24	13116	0.0000	0.0000	0.0000*	1.0000	0.0000
25	1049	0.0067	0.0113	0.0084	2.0000	0.0053
26	12315	0.0111	0.0077	0.0091	2.0000	0.0057
27	13348	0.0000	0.0000	0.0000*	1.0000	0.0000
28	10111	0.0000	0.0000	0.0000*	1.0000	0.0000
29	4271	0.0000	0.0000	0.0000*	1.0000	0.0000
30	8125	0.0064	0.0173	0.0093	2.0000	0.0059
31	2878	0.0000	0.0000	0.0000*	1.0000	0.0000
32	5226	0.0000	0.0000	0.0000*	1.0000	0.0000
33	13120	0.0000	0.0000	0.0000*	1.0000	0.0000
34	2152	0.0000	0.0000	0.0000*	1.0000	0.0000
35	3863	0.0098	0.0119	0.0107	2.0000	0.0068
36	867	0.0000	0.0000	0.0000*	1.0000	0.0000
37	3423	0.0000	0.0000	0.0000*	1.0000	0.0000
38	4636	0.0075	0.0127	0.0094	2.0000	0.0060
39	4516	0.0121	0.0133	0.0127	1.0000	0.0127
40	12152	0.0000	0.0000	0.0000*	1.0000	0.0000
41	4721	0.0000	0.0000	0.0000*	1.0000	0.0000
42	1333	0.0000	0.0000	0.0000*	1.0000	0.0000
43	8716	0.0000	0.0000	0.0000*	1.0000	0.0000
44	4298	0.0000	0.0000	0.0000*	1.0000	0.0000
45	14249	0.0000	0.0000	0.0000*	1.0000	0.0000
46	12084	0.0000	0.0000	0.0000*	1.0000	0.0000
47	787	0.0000	0.0000	0.0000*	1.0000	0.0000
48	469	0.0000	0.0000	0.0000*	1.0000	0.0000
49	8433	0.0000	0.0000	0.0000*	1.0000	0.0000

* It is counted as 0 since precision and recall values are also 0.

No	Document No	Precision	Recall	F-Measure	Granularity	Overall
50	11773	0.0000	0.0000	0.0000*	1.0000	0.0000
51	6495	0.0000	0.0000	0.0000*	1.0000	0.0000
52	4697	0.0016	0.0130	0.0028	2.0000	0.0018
53	5744	0.0000	0.0000	0.0000*	1.0000	0.0000
54	4488	0.0000	0.0000	0.0000*	1.0000	0.0000
55	5527	0.0000	0.0000	0.0000*	1.0000	0.0000
56	3926	0.0000	0.0000	0.0000*	1.0000	0.0000
57	217	0.0000	0.0000	0.0000*	1.0000	0.0000
58	502	0.0000	0.0000	0.0000*	1.0000	0.0000
59	12262	0.0000	0.0000	0.0000*	1.0000	0.0000
60	11586	0.0000	0.0000	0.0000*	1.0000	0.0000
61	13010	0.0000	0.0000	0.0000*	1.0000	0.0000
62	13393	0.0000	0.0000	0.0000*	1.0000	0.0000
63	12255	0.0000	0.0000	0.0000*	1.0000	0.0000
64	5207	0.0000	0.0000	0.0000*	1.0000	0.0000
65	8229	0.0000	0.0000	0.0000*	1.0000	0.0000
66	6507	0.0000	0.0000	0.0000*	1.0000	0.0000
67	4866	0.0000	0.0000	0.0000*	1.0000	0.0000
68	14008	0.0000	0.0000	0.0000*	1.0000	0.0000
69	3445	0.0000	0.0000	0.0000*	1.0000	0.0000
70	4807	0.0000	0.0000	0.0000*	1.0000	0.0000
71	1844	0.0000	0.0000	0.0000*	1.0000	0.0000
72	10317	0.0000	0.0000	0.0000*	1.0000	0.0000
73	2318	0.0000	0.0000	0.0000*	1.0000	0.0000
74	12707	0.0000	0.0000	0.0000*	1.0000	0.0000
75	6552	0.0000	0.0000	0.0000*	1.0000	0.0000
76	14271	0.0000	0.0000	0.0000*	1.0000	0.0000
77	4782	0.0000	0.0000	0.0000*	1.0000	0.0000
78	1536	0.0000	0.0000	0.0000*	1.0000	0.0000
79	5057	0.0000	0.0000	0.0000*	1.0000	0.0000
80	744	0.0000	0.0000	0.0000*	1.0000	0.0000
81	4858	0.0000	0.0000	0.0000*	1.0000	0.0000
82	10388	0.0000	0.0000	0.0000*	1.0000	0.0000
83	1387	0.0000	0.0000	0.0000*	1.0000	0.0000
84	4445	0.0094	0.0084	0.0089	1.0000	0.0089
85	3514	0.0000	0.0000	0.0000*	1.0000	0.0000
86	13023	0.0000	0.0000	0.0000*	1.0000	0.0000
87	5129	0.0000	0.0000	0.0000*	1.0000	0.0000
88	2100	0.0000	0.0000	0.0000*	1.0000	0.0000
89	12566	0.0000	0.0000	0.0000*	1.0000	0.0000
90	12655	0.0000	0.0000	0.0000*	1.0000	0.0000
91	117	0.0000	0.0000	0.0000*	1.0000	0.0000
92	4256	0.0000	0.0000	0.0000*	1.0000	0.0000
93	8529	0.0030	0.0144	0.0050	1.0000	0.0050
94	14244	0.0000	0.0000	0.0000*	1.0000	0.0000
95	6191	0.0000	0.0000	0.0000*	1.0000	0.0000
96	12192	0.0000	0.0000	0.0000*	1.0000	0.0000
97	2943	0.0000	0.0000	0.0000*	1.0000	0.0000
98	3072	0.0000	0.0000	0.0000*	1.0000	0.0000
99	5117	0.0000	0.0000	0.0000*	1.0000	0.0000
100	8408	0.0000	0.0000	0.0000*	1.0000	0.0000
	Average	0.0014	0.0018	0.0016	1.1200	0.0015

* It is counted as 0 since precision and recall values are also 0.

Table A.6: Levenstein metric's no obfuscation plagiarism results for randomly selected 100 documents used in the experiments.

No	Document No	Precision	Recall	F-Measure	Granularity	Overall
1	3333	0.7251	0.6208	0.6689	1.0000	0.6689
2	3369	0.5845	0.9293	0.7176	1.0000	0.7176
3	6112	1.0000	1.0000	1.0000	1.0000	1.0000
4	3976	0.7242	0.9145	0.8083	1.0000	0.8083
5	5205	0.8942	1.0000	0.9441	1.0000	0.9441
6	5567	0.6730	0.7303	0.7005	1.0000	0.7005
7	5768	1.0000	0.7950	0.8858	2.0000	0.5589
8	11524	1.0000	0.7282	0.8427	1.0000	0.8427
9	2925	0.8602	1.0000	0.9248	1.0000	0.9248
10	3962	0.8537	0.6872	0.7615	1.0000	0.7615
11	10739	0.9525	0.7825	0.8592	1.0000	0.8592
12	2956	0.8326	0.6957	0.7580	1.0000	0.7580
13	5149	0.9387	0.6467	0.7658	1.0000	0.7658
14	4216	0.9595	0.9075	0.9328	1.0000	0.9328
15	4	0.6838	0.2270	0.3408	1.0000	0.3408
16	2233	0.6556	0.8034	0.7220	1.0000	0.7220
17	13160	0.9844	0.9265	0.9545	2.0000	0.6023
18	2772	0.0000	0.0000	0.0000*	1.0000	0.0000
19	4667	0.7774	1.0000	0.8748	1.0000	0.8748
20	10081	0.7648	0.4596	0.5742	2.0000	0.3623
21	14360	1.0000	1.0000	1.0000	1.0000	1.0000
22	11926	0.9687	0.5475	0.6996	1.0000	0.6996
23	3711	0.9430	1.0000	0.9707	1.0000	0.9707
24	12402	0.7574	0.6065	0.6736	1.0000	0.6736
25	10026	0.9431	0.7688	0.8471	1.0000	0.8471
26	2175	0.7765	0.7564	0.7663	1.0000	0.7663
27	12094	0.9580	0.9692	0.9636	1.0000	0.9636
28	190	0.9767	0.8936	0.9333	1.0000	0.9333
29	13450	0.7062	0.7419	0.7236	1.0000	0.7236
30	761	0.5585	0.5996	0.5783	1.0000	0.5783
31	4693	0.8665	0.9643	0.9128	1.0000	0.9128
32	5654	0.8661	0.7089	0.7797	1.0000	0.7797
33	3828	0.9359	0.5731	0.7109	2.0000	0.4485
34	8385	0.8665	0.1042	0.1860	1.0000	0.1860
35	5690	0.9012	1.0000	0.9480	1.0000	0.9480
36	3398	0.6533	0.8026	0.7203	2.0000	0.4545
37	3996	0.8145	0.7955	0.8049	1.0000	0.8049
38	553	0.6745	1.0000	0.8056	1.0000	0.8056
39	6014	0.7619	1.0000	0.8649	1.0000	0.8649
40	3965	1.0000	1.0000	1.0000	1.0000	1.0000
41	12559	0.6267	0.7687	0.6905	1.0000	0.6905
42	8368	0.5510	0.6619	0.6014	1.0000	0.6014
43	122	0.7354	1.0000	0.8475	1.0000	0.8475
44	1468	0.7954	0.6360	0.7068	1.0000	0.7068
45	10420	0.8069	0.8205	0.8136	1.0000	0.8136
46	11905	0.9222	0.1134	0.2020	1.0000	0.2020
47	6384	1.0000	0.4338	0.6051	1.0000	0.6051
48	6560	0.9274	0.5871	0.7190	1.0000	0.7190
49	6267	0.5804	0.5084	0.5420	1.0000	0.5420

* It is counted as 0 since precision and recall values are also 0.

No	Document No	Precision	Recall	F-Measure	Granularity	Overall
50	3477	0.9169	0.7254	0.8100	1.0000	0.8100
51	13157	0.6718	0.9769	0.7961	1.0000	0.7961
52	14050	0.8023	0.6613	0.7250	1.0000	0.7250
53	6341	0.8115	0.7676	0.7889	1.0000	0.7889
54	576	0.6606	0.9744	0.7874	2.0000	0.4968
55	10408	0.8697	0.5278	0.6569	1.0000	0.6569
56	972	0.8890	0.7225	0.7971	1.0000	0.7971
57	14235	0.5784	0.9696	0.7246	1.0000	0.7246
58	13303	0.8686	0.8348	0.8514	1.0000	0.8514
59	2344	0.7822	0.8160	0.7987	1.0000	0.7987
60	14358	0.8458	0.5211	0.6449	1.0000	0.6449
61	8248	0.6932	0.6099	0.6489	2.0000	0.4094
62	3097	0.9103	1.0000	0.9530	1.0000	0.9530
63	12334	0.8252	1.0000	0.9042	1.0000	0.9042
64	1988	0.9890	0.7242	0.8361	1.0000	0.8361
65	2705	0.9506	0.7910	0.8635	1.0000	0.8635
66	917	0.7826	0.8090	0.7956	1.0000	0.7956
67	6575	0.7499	0.4881	0.5913	1.0000	0.5913
68	1729	0.8252	0.6475	0.7256	1.0000	0.7256
69	1166	0.8170	0.8567	0.8364	1.0000	0.8364
70	14401	0.8347	0.5446	0.6591	1.0000	0.6591
71	6194	0.7773	0.7637	0.7704	1.0000	0.7704
72	6026	0.6546	0.9442	0.7732	1.0000	0.7732
73	2025	0.8127	0.9043	0.8561	1.0000	0.8561
74	8564	0.6125	0.7570	0.6771	1.0000	0.6771
75	12044	0.8151	0.7387	0.7750	1.0000	0.7750
76	2327	0.8104	1.0000	0.8953	1.0000	0.8953
77	4982	0.7093	0.3941	0.5067	1.0000	0.5067
78	3011	0.5430	0.8561	0.6645	1.0000	0.6645
79	1469	0.7927	0.8884	0.8378	1.0000	0.8378
80	1373	0.6650	1.0000	0.7988	2.0000	0.5040
81	4658	0.9151	0.8850	0.8998	1.0000	0.8998
82	5701	0.8041	0.7865	0.7952	2.0000	0.5017
83	2725	0.6126	0.5184	0.5616	1.0000	0.5616
84	3406	0.7394	0.8461	0.7892	2.0000	0.4979
85	11927	0.8721	0.8369	0.8541	1.0000	0.8541
86	14231	0.5287	0.9766	0.6860	1.0000	0.6860
87	1985	0.6451	0.9320	0.7625	1.0000	0.7625
88	1135	0.6666	0.1000	0.1739	1.0000	0.1739
89	1606	0.7007	0.7765	0.7367	2.0000	0.4648
90	8622	0.9313	0.5996	0.7295	1.0000	0.7295
91	11720	0.8831	1.0000	0.9379	1.0000	0.9379
92	135	0.9570	0.5285	0.6809	2.0000	0.4296
93	6695	0.7300	0.9270	0.8168	1.0000	0.8168
94	6209	0.5454	0.7801	0.6420	1.0000	0.6420
95	3785	0.7828	1.0000	0.8782	1.0000	0.8782
96	4257	0.8076	0.9377	0.8678	1.0000	0.8678
97	6512	0.8239	0.3004	0.4403	1.0000	0.4403
98	12024	1.0000	0.0800	0.1481	1.0000	0.1481
99	12530	0.9890	0.7643	0.8623	1.0000	0.8623
100	6227	0.7394	0.9375	0.8267	1.0000	0.8267
	Average	0.7968	0.7401	0.7674	1.1200	0.7079

Table A.7: Levenstein metric's low plagiarism results for randomly selected 100 documents used in the experiments.

No	Document No	Precision	Recall	F-Measure	Granularity	Overall
1	4610	0.1175	0.0282	0.0455	2.0000	0.0287
2	825	0.0188	0.0138	0.0159	2.0000	0.0100
3	6103	0.0374	0.0312	0.0340	2.0000	0.0215
4	3740	0.0000	0.0000	0.0000*	1.0000	0.0000
5	2967	0.0433	0.0322	0.0369	2.0000	0.0233
6	6626	0.0856	0.0141	0.0242	2.0000	0.0153
7	6158	0.0695	0.0287	0.0406	2.0000	0.0256
8	8602	0.0455	0.0293	0.0356	2.0000	0.0225
9	3967	0.0798	0.0192	0.0310	1.0000	0.0310
10	5007	0.0000	0.0000	0.0000*	1.0000	0.0000
11	1851	0.0589	0.0218	0.0318	2.0000	0.0201
12	2637	0.0636	0.0122	0.0205	2.0000	0.0129
13	5513	0.0504	0.0076	0.0132	2.0000	0.0083
14	2865	0.0290	0.0156	0.0203	2.0000	0.0128
15	13425	0.0328	0.0247	0.0282	1.0000	0.0282
16	8384	0.0000	0.0000	0.0000*	1.0000	0.0000
17	8871	0.0952	0.0204	0.0336	2.0000	0.0212
18	1062	0.0457	0.0268	0.0338	2.0000	0.0213
19	3512	0.0543	0.0266	0.0357	2.0000	0.0225
20	1829	0.0000	0.0000	0.0000*	1.0000	0.0000
21	4569	0.0397	0.0231	0.0292	2.0000	0.0184
22	382	0.0147	0.0210	0.0173	1.0000	0.0173
23	12048	0.0753	0.0193	0.0307	2.0000	0.0194
24	12118	0.0440	0.0097	0.0159	1.0000	0.0159
25	5872	0.0522	0.0157	0.0241	2.0000	0.0152
26	1086	0.0176	0.0163	0.0169	1.0000	0.0169
27	8747	0.0000	0.0000	0.0000*	1.0000	0.0000
28	3752	0.0561	0.0272	0.0366	2.0000	0.0231
29	2857	0.0458	0.0231	0.0307	2.0000	0.0194
30	1751	0.0466	0.0254	0.0329	2.0000	0.0207
31	8046	0.0516	0.0251	0.0338	2.0000	0.0213
32	2416	0.0946	0.0273	0.0424	2.0000	0.0267
33	3980	0.0347	0.0264	0.0300	2.0000	0.0189
34	3820	0.0106	0.0206	0.0140	2.0000	0.0088
35	8323	0.0000	0.0000	0.0000*	1.0000	0.0000
36	11949	0.0000	0.0000	0.0000*	1.0000	0.0000
37	3898	0.0705	0.0189	0.0298	2.0000	0.0188
38	13199	0.0242	0.0298	0.0267	2.0000	0.0169
39	2791	0.0173	0.0265	0.0209	2.0000	0.0132
40	14135	0.0581	0.0152	0.0241	2.0000	0.0152
41	4162	0.0400	0.0227	0.0290	2.0000	0.0183
42	3771	0.0000	0.0000	0.0000*	1.0000	0.0000
43	2447	0.0145	0.0318	0.0199	2.0000	0.0126
44	14342	0.0706	0.0166	0.0269	2.0000	0.0170
45	10657	0.0000	0.0000	0.0000*	1.0000	0.0000
46	3859	0.0092	0.0286	0.0139	2.0000	0.0088
47	260	0.1101	0.0324	0.0501	1.0000	0.0501
48	8187	0.0000	0.0000	0.0000*	1.0000	0.0000
49	14133	0.0953	0.0262	0.0411	2.0000	0.0259

* It is counted as 0 since precision and recall values are also 0.

No	Document No	Precision	Recall	F-Measure	Granularity	Overall
50	2451	0.0627	0.0149	0.0241	1.0000	0.0241
51	2068	0.0637	0.0066	0.0120	2.0000	0.0075
52	968	0.0857	0.0171	0.0285	2.0000	0.0180
53	1850	0.0000	0.0000	0.0000*	1.0000	0.0000
54	4285	0.0599	0.0245	0.0348	2.0000	0.0219
55	3751	0.0000	0.0000	0.0000*	1.0000	0.0000
56	1541	0.0517	0.0209	0.0298	2.0000	0.0188
57	5764	0.0589	0.0187	0.0284	1.0000	0.0284
58	4430	0.0394	0.0216	0.0279	1.0000	0.0279
59	14367	0.0225	0.0102	0.0140	2.0000	0.0089
60	2173	0.0154	0.0076	0.0102	2.0000	0.0064
61	10031	0.0479	0.0106	0.0174	2.0000	0.0110
62	434	0.0880	0.0175	0.0292	1.0000	0.0292
63	4904	0.0759	0.0162	0.0267	2.0000	0.0168
64	111	0.1276	0.0162	0.0287	1.0000	0.0287
65	11921	0.0527	0.0061	0.0109	2.0000	0.0069
66	3044	0.0361	0.0275	0.0312	1.0000	0.0312
67	4668	0.0500	0.0154	0.0235	2.0000	0.0149
68	12056	0.0000	0.0000	0.0000*	1.0000	0.0000
69	460	0.0688	0.0464	0.0554	1.0000	0.0554
70	8504	0.0970	0.0229	0.0371	2.0000	0.0234
71	13386	0.0719	0.0069	0.0126	1.0000	0.0126
72	925	0.0000	0.0000	0.0000*	1.0000	0.0000
73	3297	0.0470	0.0283	0.0353	2.0000	0.0223
74	5109	0.1235	0.0146	0.0261	2.0000	0.0165
75	13235	0.0000	0.0000	0.0000*	1.0000	0.0000
76	6177	0.0847	0.0170	0.0283	2.0000	0.0179
77	8968	0.0517	0.0165	0.0250	1.0000	0.0250
78	14142	0.0000	0.0000	0.0000*	1.0000	0.0000
79	10553	0.0528	0.0275	0.0362	2.0000	0.0228
80	11867	0.0409	0.0179	0.0249	1.0000	0.0249
81	13093	0.0658	0.0255	0.0368	1.0000	0.0368
82	6171	0.0443	0.0271	0.0336	1.0000	0.0336
83	3592	0.0745	0.0217	0.0336	1.0000	0.0336
84	8991	0.0572	0.0158	0.0248	2.0000	0.0156
85	3776	0.0472	0.0268	0.0342	2.0000	0.0216
86	1267	0.0743	0.0205	0.0321	2.0000	0.0203
87	8406	0.0269	0.0117	0.0163	2.0000	0.0103
88	6006	0.0000	0.0000	0.0000*	1.0000	0.0000
89	198	0.0000	0.0000	0.0000*	1.0000	0.0000
90	5038	0.0261	0.0308	0.0283	2.0000	0.0178
91	6687	0.0000	0.0000	0.0000*	1.0000	0.0000
92	2969	0.0000	0.0000	0.0000*	1.0000	0.0000
93	13200	0.0187	0.0169	0.0178	1.0000	0.0178
94	8039	0.0708	0.0127	0.0215	2.0000	0.0136
95	3025	0.0806	0.0228	0.0355	2.0000	0.0224
96	8356	0.0576	0.0131	0.0213	1.0000	0.0213
97	6362	0.0257	0.0169	0.0204	2.0000	0.0129
98	3912	0.0000	0.0000	0.0000*	1.0000	0.0000
99	1443	0.0000	0.0000	0.0000*	1.0000	0.0000
100	294	0.0285	0.0161	0.0206	2.0000	0.0130
	Average	0.0430	0.0161	0.0234	1.5700	0.0172

* It is counted as 0 since precision and recall values are also 0.

Table A.8: Levenstein metric's high plagiarism results for randomly selected 100 documents used in the experiments.

No	Document No	Precision	Recall	F-Measure	Granularity	Overall
1	10608	0.0000	0.0000	0.0000*	1.0000	0.0000
2	10398	0.0000	0.0000	0.0000*	1.0000	0.0000
3	2345	0.0080	0.0079	0.0079	2.0000	0.0050
4	8870	0.0060	0.0095	0.0074	2.0000	0.0046
5	5642	0.0069	0.0070	0.0069	1.0000	0.0069
6	10617	0.0000	0.0000	0.0000*	1.0000	0.0000
7	4095	0.0000	0.0000	0.0000*	1.0000	0.0000
8	6207	0.0000	0.0000	0.0000*	1.0000	0.0000
9	12023	0.0000	0.0000	0.0000*	1.0000	0.0000
10	6569	0.0079	0.0082	0.0080	2.0000	0.0051
11	2289	0.0000	0.0000	0.0000*	1.0000	0.0000
12	1853	0.0000	0.0000	0.0000*	1.0000	0.0000
13	14290	0.0000	0.0000	0.0000*	1.0000	0.0000
14	919	0.0000	0.0000	0.0000*	1.0000	0.0000
15	3639	0.0000	0.0000	0.0000*	1.0000	0.0000
16	2443	0.0000	0.0000	0.0000*	1.0000	0.0000
17	3853	0.0064	0.0067	0.0065	2.0000	0.0041
18	458	0.0000	0.0000	0.0000*	1.0000	0.0000
19	6497	0.0000	0.0000	0.0000*	1.0000	0.0000
20	5089	0.0000	0.0000	0.0000*	1.0000	0.0000
21	13359	0.0000	0.0000	0.0000*	1.0000	0.0000
22	2054	0.0000	0.0000	0.0000*	1.0000	0.0000
23	12263	0.0000	0.0000	0.0000*	1.0000	0.0000
24	13116	0.0000	0.0000	0.0000*	1.0000	0.0000
25	1049	0.0074	0.0072	0.0073	2.0000	0.0046
26	12315	0.0000	0.0000	0.0000*	1.0000	0.0000
27	13348	0.0000	0.0000	0.0000*	1.0000	0.0000
28	10111	0.0000	0.0000	0.0000*	1.0000	0.0000
29	4271	0.0000	0.0000	0.0000*	1.0000	0.0000
30	8125	0.0021	0.0095	0.0034	2.0000	0.0022
31	2878	0.0000	0.0000	0.0000*	1.0000	0.0000
32	5226	0.0000	0.0000	0.0000*	1.0000	0.0000
33	13120	0.0000	0.0000	0.0000*	1.0000	0.0000
34	2152	0.0000	0.0000	0.0000*	1.0000	0.0000
35	3863	0.0111	0.0088	0.0098	2.0000	0.0062
36	867	0.0000	0.0000	0.0000*	1.0000	0.0000
37	3423	0.0000	0.0000	0.0000*	1.0000	0.0000
38	4636	0.0052	0.0056	0.0054	2.0000	0.0034
39	4516	0.0018	0.0085	0.0030	2.0000	0.0019
40	12152	0.0000	0.0000	0.0000*	1.0000	0.0000
41	4721	0.0000	0.0000	0.0000*	1.0000	0.0000
42	1333	0.0000	0.0000	0.0000*	1.0000	0.0000
43	8716	0.0000	0.0000	0.0000*	1.0000	0.0000
44	4298	0.0000	0.0000	0.0000*	1.0000	0.0000
45	14249	0.0000	0.0000	0.0000*	1.0000	0.0000
46	12084	0.0000	0.0000	0.0000*	1.0000	0.0000
47	787	0.0000	0.0000	0.0000*	1.0000	0.0000
48	469	0.0000	0.0000	0.0000*	1.0000	0.0000
49	8433	0.0000	0.0000	0.0000*	1.0000	0.0000

* It is counted as 0 since precision and recall values are also 0.

No	Document No	Precision	Recall	F-Measure	Granularity	Overall
50	11773	0.0000	0.0000	0.0000*	1.0000	0.0000
51	6495	0.0000	0.0000	0.0000*	1.0000	0.0000
52	4697	0.0046	0.0085	0.0060	2.0000	0.0038
53	5744	0.0000	0.0000	0.0000*	1.0000	0.0000
54	4488	0.0000	0.0000	0.0000*	1.0000	0.0000
55	5527	0.0000	0.0000	0.0000*	1.0000	0.0000
56	3926	0.0000	0.0000	0.0000*	1.0000	0.0000
57	217	0.0000	0.0000	0.0000*	1.0000	0.0000
58	502	0.0000	0.0000	0.0000*	1.0000	0.0000
59	12262	0.0000	0.0000	0.0000*	1.0000	0.0000
60	11586	0.0000	0.0000	0.0000*	1.0000	0.0000
61	13010	0.0000	0.0000	0.0000*	1.0000	0.0000
62	13393	0.0000	0.0000	0.0000*	1.0000	0.0000
63	12255	0.0000	0.0000	0.0000*	1.0000	0.0000
64	5207	0.0000	0.0000	0.0000*	1.0000	0.0000
65	8229	0.0000	0.0000	0.0000*	1.0000	0.0000
66	6507	0.0000	0.0000	0.0000*	1.0000	0.0000
67	4866	0.0000	0.0000	0.0000*	1.0000	0.0000
68	14008	0.0000	0.0000	0.0000*	1.0000	0.0000
69	3445	0.0000	0.0000	0.0000*	1.0000	0.0000
70	4807	0.0000	0.0000	0.0000*	1.0000	0.0000
71	1844	0.0000	0.0000	0.0000*	1.0000	0.0000
72	10317	0.0000	0.0000	0.0000*	1.0000	0.0000
73	2318	0.0000	0.0000	0.0000*	1.0000	0.0000
74	12707	0.0000	0.0000	0.0000*	1.0000	0.0000
75	6552	0.0000	0.0000	0.0000*	1.0000	0.0000
76	14271	0.0000	0.0000	0.0000*	1.0000	0.0000
77	4782	0.0000	0.0000	0.0000*	1.0000	0.0000
78	1536	0.0000	0.0000	0.0000*	1.0000	0.0000
79	5057	0.0000	0.0000	0.0000*	1.0000	0.0000
80	744	0.0000	0.0000	0.0000*	1.0000	0.0000
81	4858	0.0000	0.0000	0.0000*	1.0000	0.0000
82	10388	0.0000	0.0000	0.0000*	1.0000	0.0000
83	1387	0.0000	0.0000	0.0000*	1.0000	0.0000
84	4445	0.0055	0.0077	0.0064	2.0000	0.0040
85	3514	0.0000	0.0000	0.0000*	1.0000	0.0000
86	13023	0.0000	0.0000	0.0000*	1.0000	0.0000
87	5129	0.0000	0.0000	0.0000*	1.0000	0.0000
88	2100	0.0000	0.0000	0.0000*	1.0000	0.0000
89	12566	0.0000	0.0000	0.0000*	1.0000	0.0000
90	12655	0.0000	0.0000	0.0000*	1.0000	0.0000
91	117	0.0000	0.0000	0.0000*	1.0000	0.0000
92	4256	0.0000	0.0000	0.0000*	1.0000	0.0000
93	8529	0.0058	0.0067	0.0062	2.0000	0.0039
94	14244	0.0000	0.0000	0.0000*	1.0000	0.0000
95	6191	0.0000	0.0000	0.0000*	1.0000	0.0000
96	12192	0.0000	0.0000	0.0000*	1.0000	0.0000
97	2943	0.0000	0.0000	0.0000*	1.0000	0.0000
98	3072	0.0000	0.0000	0.0000*	1.0000	0.0000
99	5117	0.0000	0.0000	0.0000*	1.0000	0.0000
100	8408	0.0000	0.0000	0.0000*	1.0000	0.0000
	Average	0.0008	0.0010	0.0009	1.1200	0.0008

* It is counted as 0 since precision and recall values are also 0.